STUDI INDEPENDEN MACHINE LEARNING BANGKIT 2023: ANALISIS DEEP LEARNING UNTUK MASALAH KESEHATAN MENTAL

KERJA PRAKTIK

Diajukan untuk Memenuhi Persyaratan Akademik dalam

Menyelesaikan Pendidikan pada Program Studi

S1 Teknik Informatika Universitas Kristen Maranatha

Oleh

Sherly Santiadi

2072025

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS KRISTEN MARANATHA

BANDUNG

2023

LEMBAR PENGESAHAN

STUDI INDEPENDEN MACHINE LEARNING BANGKIT 2023: ANALISIS DEEP LEARNING UNTUK MASALAH KESEHATAN MENTAL

Dengan ini, saya menyatakan bahwa
isi CD ROM Laporan Penelitian sama dengan hasil revisi akhir

Bandung, 22 Mei 2023

(Sherly Santiadi)
(2072025)

Menyetujui,
Pembimbing I

Prof. Dr. Ir. Mewati Ayub, M.T.
NIK: 720140

Penguji I

Oscar Karnalim, Ph.D.
NIK: 720309

Mengetahui,
Ketua Program Studi Teknik Informatika

Julianti Kasih, S.E., M.Kom.
NIK: 720286

PERNYATAAN ORISINALITAS LAPORAN PENELITIAN

Dengan ini, saya yang bertanda tangan di bawah ini:

Nama	:	Sherly Santiadi
NRP	:	2072025
Fakultas/ Program Studi	:	Teknologi Informasi/ Teknik Informatika

Menyatakan bahwa laporan penelitian ini adalah benar merupakan hasil karya saya sendiri dan bukan duplikasi dari orang lain.

Apabila pada masa mendatang diketahui bahwa pernyataan ini tidak benar adanya, saya bersedia menerima sanksi yang diberikan dengan segala konsekuensinya.

Demikian pernyataan ini saya buat.

Bandung, 22 Mei 2023

Sherly Santiadi
NRP: 2072025

PERNYATAAN PUBLIKASI LAPORAN PENELITIAN

Saya yang bertanda tangan di bawah ini:

Nama	:	Sherly Santiadi
NRP	:	2072025
Fakultas/ Program Studi	:	Teknologi Informasi/ Teknik Informatika

Dengan ini, saya menyatakan bahwa:

1. Demi perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan kepada Universitas Kristen Maranatha Hak Bebas Royalti non eksklusif (*Non Exclusive Royalty Free Right*) atas laporan penelitian saya yang berjudul STUDI INDEPENDEN MACHINE LEARNING BANGKIT 2023: ANALISIS DEEP LEARNING UNTUK MASALAH KESEHATAN MENTAL
2. Universitas Kristen Maranatha Bandung berhak menyimpan, mengalihmediakan/ mengalihformatkan, mengelola dalam bentuk pangkalan data (*database*), mendistribusikannya, serta menampilkannya dalam bentuk *softcopy* untuk kepentingan akademis tanpa perlu meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/ pencipta.
3. Saya bersedia dan menjamin untuk menanggung secara pribadi tanpa melibatkan pihak Universitas Kristen Maranatha Bandung, segala bentuk tuntutan hukum yang timbul atas pelanggaran hak cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya dan untuk dapat dipergunakan sebagaimana mestinya.

Bandung, 22 Mei 2023

Sherly Santiadi

NRP: 2072025

PRAKATA

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa, atas segala rahmat dan karunia-Nya, sehingga laporan kerja praktik yang berjudul "Studi Independen Machine Learning Bangkit 2023: Analisis *Deep Learning* untuk Masalah Kesehatan Mental" dapat diselesaikan dengan tepat waktu. Adapun tujuan penulisan laporan kerja praktik ini yaitu sebagai salah syarat kelulusan dalam Mata Kuliah IN270 Kerja Praktik di Fakultas Teknologi Informasi Program Studi S1 Teknik Informatika.

Oleh karena itu, penulis ingin mengucapkan terima kasih kepada seluruh pihak yang terlibat khususnya kepada Ibu Prof. Dr. Ir. Mewati Ayub, M.T. selaku dosen pembimbing yang telah memberikan banyak pengarahan dalam penulisan laporan kerja praktik, serta segala bentuk kritik maupun saran yang membangun. Selain itu juga, penulis ingin mengucapkan terima kasih kepada:

1. Bapak Ir. Teddy Marcus Zakaria, M.T. selaku Dekan Fakultas Teknologi Informasi.
2. Ibu Julianti Kasih, S.E., M.Kom. selaku Ketua Program Studi Teknik Informatika.
3. Bapak Sulaeman Santoso, S.Kom., M.T. selaku Koordinator Kerja Praktik.
4. Bapak Hendra Bunyamin, S.Si., M.T. selaku Dosen Kecerdasan Mesin.
5. Keluarga dan rekan-rekan.

Penulis menyadari bahwa laporan kerja praktik ini masih memiliki banyak kekurangan, oleh karena itu penulis mengharapkan untuk mendapatkan kritik dan saran yang membangun dari para pembaca. Laporan kerja praktik ini diharapkan dapat dijadikan sebagai sebuah referensi maupun wawasan baru bagi para pembaca.

Bandung, 22 Mei 2023

Sherly Santiadi

ABSTRAK

Analisis sentimen merupakan sebuah teknik pembelajaran mesin yang bertujuan untuk mengekstrak informasi subjektif dari sebuah teks. Dalam penelitian ini, analisis sentimen digunakan untuk menganalisis jurnal yang diisi oleh pengguna untuk menentukan probabilitas pengguna mengalami depresi atau tidak. Sistem yang diusulkan dalam laporan ini menggunakan teknik pemrosesan bahasa alami. Terdapat dua model yang dikembangkan di dalam laporan ini yaitu model LSTM dan model yang sudah diimprovisasi yaitu Bidirectional LSTM. Untuk mengetahui kinerja model, maka di akhir percobaan kinerja model diukur menggunakan metriks akurasi. Hasil percobaan yang telah dilakukan menunjukkan bahwa sistem yang diusulkan cukup efektif dalam mengidentifikasi sentimen jurnal harian dan memprediksi kemungkinan pengguna mengalami depresi atau tidak. Selain itu, terdapat juga fitur lainnya yang tidak dibahas penuh di dalam laporan ini namun secara keseluruhan, sistem yang diusulkan merupakan aplikasi yang berguna untuk mencegah, memantau, dan mengatasi keadaan emosional dari penggunanya. Hal ini dapat digunakan untuk memberikan deteksi dini pada gejala depresi dan masalah kesehatan mental lainnya, yang dapat membantu dalam memberikan perawatan tepat waktu untuk meningkatkan kualitas hidup bagi penggunanya.

Kata kunci: aplikasi kesehatan mental, pembelajaran mesin, pemrosesan bahasa alami.

ABSTRACT

Sentiment analysis is a machine learning technique for extracting subjective information from text. In this study, sentiment analysis was utilized to examine user journals to determine whether or not the user experienced depression. Natural language processing techniques are used in the system suggested in this research. There are two models developed in this report, first is the LSTM model and an improvised model which is Bidirectional LSTM. To find out the performance model, at the end of the experimental model, the performance is measured using an accuracy metrics. The results of the studies conducted reveal that the suggested approach is highly good in recognizing sentiment daily journals and predicting whether or not users are depressed. There are other features that are not completely described in this report, but overall, the suggested system is an application that is effective for preventing, monitoring, and overcoming with its users' emotional states. It can be used to detect indications of depression and other mental health issues early, which can aid in giving prompt treatment and improving users' quality of life.

Keywords: machine learning, mental health applications, natural language processing.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR SINGKATAN

API	Application Programming Interface
HDF	Hierarchical Data Format
IT	Information Technology
LSTM	Long Short-Term Memory
PCA	Principal Component Analysis
SQL	Structured Query Language
UML	Unified Modeling Language

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Program Bangkit 2023 merupakan salah satu program Kampus Merdeka yang diselenggarakan oleh Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi yaitu Bapak Nadiem Makarim pada bulan Januari tahun 2020. Tujuan dari diadakannya Program Kampus Merdeka antara lain ialah agar dapat membantu mahasiswa dalam memperkaya keilmuan agar siap menghadapi dunia kerja. Pembelajarannya yang kaya akan permasalahan nyata di dalam dunia industri akan membantu mahasiswa untuk mengasah kemampuan, pola pikir, manajemen diri, serta interaksi sosial. Bentuk-bentuk kegiatan yang ada di dalam Program Kampus Merdeka ini meliputi Pertukaran Pelajar, Magang/Praktik Kerja, Asistensi Mengajar di Satuan Pendidikan, Penelitian/Riset, Proyek Kemanusiaan, Kegiatan Wirausaha, Studi/Proyek Independen, dan Membangun Desa/Kuliah Kerja Nyata Tematik [1].

Dari berbagai bentuk kegiatan yang diselenggarakan dalam Program Kampus Merdeka, mahasiswa diperkenankan untuk memilih satu program yang akan diikuti dalam waktu satu semester. Di dalam banyaknya program tersebut, terdapat mitra-mitra yang siap memberikan wadah bagi mahasiswa dalam mengembangkan potensi dirinya, salah satunya adalah PT. Dicoding Akademi Indonesia. PT. Dicoding Akademi Indonesia merupakan sebuah perusahaan berbasis platform edukasi teknologi yang membantu menjembatani pelajar untuk mengembangkan potensi akademik dan *softskills* di bidang teknologi [2].

PT. Dicoding resmi menjadi salah satu mitra pada Program Kampus Merdeka sejak tahun 2020 dengan nama Program Bangkit. Program Bangkit pada awalnya hanya diikuti oleh 300 peserta dari berbagai universitas maupun jurusan. Dengan seiring meningkatnya peminat di bidang teknologi, maka pada tahun 2021 Program Bangkit memberikan 3000 kuota yang dapat diikuti oleh peserta terpilih. Pada tahun 2023, kuota yang disediakan meningkat menjadi 5000 peserta. Di samping penambahan kuota, kurikulum yang ditawarkan oleh Program Bangkit ini semakin beragam yaitu *Machine Learning*, *Android*, serta *Cloud Computing* [3].

Penambahan kuota serta kurikulum yang diterapkan oleh Bangkit sendiri dikarenakan selain oleh karena faktor meningkatnya peminat dari tahun ke tahun, terdapat faktor lain yaitu kebutuhan akan teknologi yang berkaitan erat dengan *Machine Learning*, *Android*, serta *Cloud Computing* juga meningkat. Di dalam program ini juga, terdapat projek akhir yang disebut dengan *Capstone Project* yang wajib diikuti oleh seluruh peserta Bangkit. *Capstone Project* ini terbagi ke dalam dua kategori yaitu *Company Project* maupun *Product-based Project*. Di dalam laporan ini, penulis beserta rekan lainnya yang tergabung ke dalam tim memilih *Product-based Project* yaitu dengan membangun *image captioning* berbasis aplikasi mobile untuk membantu penyandang disabilitas agar bisa lebih memahami kondisi lingkungan di sekitarnya.

1.2 Rumusan Masalah

Dalam Studi Independen Machine Learning Bangkit, terdapat beberapa modul yang perlu diselesaikan sebelum mengerjakan *capstone project*. Beberapa modul tersebut diantaranya berkaitan dengan pemrograman dasar menggunakan *Python*; melakukan analisis data menggunakan *Tableu*, *SQL*, dan *Excel*; fondasi matematika untuk *Machine Learning* seputar Aljabar Linear, Kalkulus Multivariat, dan *Principal Component Analysis*; serta pengenalan terhadap beberapa algoritma *Machine Learning* dan *Deep Learning*. Oleh karena itu, setiap permasalahan yang telah diidentifikasi dalam latar belakang akan diselesaikan dengan metode-metode yang telah didapatkan selama menempuh studi independen, dengan rumusan sebagai berikut:

1. Bagaimana hasil visualisasi data dari *dataset* yang digunakan?
2. Bagaimana cara mengimplementasikan algoritma *deep learning* dalam memantau masalah kesehatan mental?
3. Bagaimana hasil evaluasi model terhadap masalah tersebut?

1.3 Tujuan Pembahasan

Adapun tujuan pembahasan dari penulisan laporan ini, yaitu:

1. Menganalisis hasil visualisasi data dari *dataset* yang digunakan.

2. Mengimplementasikan algoritma *deep learning* dalam memantau masalah kesehatan mental.
3. Menganalisis hasil evaluasi model terhadap masalah tersebut.

1.4 Ruang Lingkup

Adapun ruang lingkup yang berisikan informasi mengenai apa saja yang akan dibahas/ dikerjakan dan apa saja yang tidak dibahas/ dikerjakan. Bagian ini bertujuan untuk membatasi apa yang akan dikerjakan dalam Laporan Studi Independen Bangkit. Batasan tersebut berupa *dataset* yang diolah, metode atau *framework* yang digunakan, detail dari batasan analisis, serta luaran. Berikut merupakan ruang lingkup dari penulisan laporan ini, yaitu:

1. Dataset yang digunakan berasal dari *Kaggle, Google Dataset,* dan pengambilan data pribadi pada tahun 2023.
2. Algoritma baseline yaitu *LSTM* dan algoritma improvisasi *Bidirectional LSTM* dibangun menggunakan pustaka *Tensorflow* dan pustaka lainnya.
3. Dataset diakses melalui *Google Cloud* menggunakan *Cloud API.*
4. Luaran yang dihasilkan berupa laporan dan aplikasi mobile (android) yang dapat diakses melalui *Google Play Store*.

1.5 Sumber Data

Sumber data yang digunakan untuk melaksanakan kerja praktik berupa data sekunder yang diperoleh dari dari *Kaggle, Google Dataset*, dan pengambilan data pribadi pada tahun 2023. Pemilihan sumber data ini bertujuan untuk efisiensi waktu, dikarenakan data tersebut akan segera dilakukan pembersihan data sebelum akhirnya dikirimkan ke dalam model *machine learning*. Selain itu, data tersebut perlu supervisi dari ahli maka akan lebih mudah mengambil data yang bersifat sekunder.

1.6 Sistematika Penyajian

Sistematika penyajian dari penyusunan laporan ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini meliputi latar belakang, rumusan masalah, tujuan pembahasan, ruang lingkup, sumber data, dan sistematika penyajian.

BAB II KAJIAN TEORI

Bab ini meliputi penjelasan singkat mengenai kajian teori yang telah didapatkan selama mengikuti Program Bangkit 2023. Selain itu penambahan teori dari berbagai sumber juga disajikan untuk melengkapi dasar teori sebelum melakukan analisis dan rancangan model.

BAB III ANALISIS DAN RANCANGAN

Bab ini meliputi analisis sistem, perancangan sistem yaitu antarmuka dengan pengguna, antarmuka perangkat keras, dan antarmuka perangkat lunak. Selain itu, di dalam bab ini juga disajikan berbagai diagram yang dibutuhkan untuk membangun aplikasi DailyCloud.

BAB IV IMPLEMENTASI

Bab ini meliputi implementasi model yang sudah dibangun berdasarkan teori-teori yang didapatkan selama mengikuti Program Bangkit 2023. Di dalam bab ini meliputi visualisasi, proses pembersihan, dan pembangunan model.

BAB V PENGUJIAN

Bab ini meliputi pengujian yang dilakukan terhadap model yang sudah dibangun. Pengujian ini dilakukan dengan dua pendekatan yaitu dengan metode *black box* dan juga *white box*.

BAB VI SIMPULAN DAN SARAN

Bab ini meliputi kesimpulan dan saran yang dapat digunakan untuk meningkatkan hasil implementasi yang telah diuji coba dalam laporan ini. Selain itu, di dalam bab ini juga terdapat hasil refleksi selama mengikuti Program Bangkit 2023.

BAB 2
KAJIAN TEORI

2.1 *IT Automation with Python*

Python merupakan sebuah bahasa pemrograman yang memiliki berbagai sintaks, *library*, dan selain itu Python juga bersifat *open-source* [4]. Dari berbagai karakteristik tersebutlah, Python menjadi bahasa pemrograman yang paling diminati oleh pekerja khususnya dalam mengotomatisasi sebuah pekerjaan yang cenderung bersifat repetitif. Akan tetapi, sebelum membangun sebuah sistem yang dapat mengotomatisasi sebuah pekerjaan, maka pekerja tersebut harus memiliki fondasi seputar logika pemrograman menggunakan Python, bagaimana Python dapat berinteraksi di dalam sistem operasi yang digunakan, menerapkan Git dan Github untuk mengelola kode, teknik-teknik dalam melakukan *troubleshooting* dan *debugging*, serta menerapkan konfigurasi Pemrograman Python menggunakan cloud.

2.1.1 *Crash Course on Python*

Crash Course on Python merupakan sebuah kursus yang dibangun oleh Google dan dapat diakses melalui *Platform Coursera*. Kursus ini terbagi ke dalam 6 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Siswa akan diajarkan terkait logika di dalam bahasa pemrograman, cara penulisan *script* yang efisien dan efektif, dan juga teori seputar otomatisasi menggunakan Python. Tidak sampai di situ saja, di dalam modul ini juga terdapat pemahaman terkait tipe data primitif dan non-primitif, logika perulangan menggunakan *for* dan *while*, serta paradigma pemrograman berorientasi objek [5].

2.1.2 *Using Python to Interact with the Operating System*

Using Python to Interact with the Operating System merupakan kursus yang memiliki 7 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti

dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan cara mempersiapkan *environment* di dalam sistem operasi kita (*Windows*/ *MacOS*/ *Linux*), memahami konsep terkait perbedaan bahasa pemrograman yang menggunakan *interpreter* ataupun *compiler* dalam mengeksekusi kode program, memahami konsep untuk melakukan manipulasi berbagai tipe *file* (*.txt, .csv,* dan lain sebagainya). Adapun konsep yang tidak kalah menarik yaitu *Regular Expression* (*Regex*) dan *bash scripting* yang ikut dibahas dalam kursus ini. *Regex* sendiri merupakan sebuah konsep yang sering digunakan di dalam *Natural Language Processing* (*NLP*) dalam melakukan pemrosesan data teks. Selain itu, siswa juga diberikan pemahaman dalam melakukan *unit testing* menggunakan *Jupyter Notebook. Unit testing* berfungsi dalam memperkecil kesalahan maupun *bug* di dalam sekumpulan kode program yang dibangun [6] .

2.1.3 *Introduction to Git and Github*

Introduction to Git and Github merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan cara menyimpan memanajemen dan mengimplementasi *rolling back* (mengembalikan kode program ke dalam versi sebelum terjadinya perubahan) sebelum adanya *tool version control* seperti Github, Bitbucket, dan lain sebagainya. Salah satu caranya adalah dengan mengutilisasi penggunaan *diff* dan *patch* untuk mengotomatisasi jika terjadinya perubahan di dalam kode program. Setelah memahami cara penggunaan *diff* dan *patch* maka siswa akan diajarkan perintah dasar Git yang biasa digunakan baik untuk penggunaan personal maupun berkelompok seperti perintah *git push, git pull, git commit, git add,* dan lain-lain [7].

2.1.4 *Troubleshooting and Debugging Techniques*

Troubleshooting and Debugging Techniques merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti

dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan cara untuk mengimplementasikan *troubleshooting* (pencarian sumber masalah dan mencoba menyelesaikan masalah tersebut) dan *debugging* (mengidentifikasi *bug* atau *error* dari sebuah kode program). Dalam mengaplikasikan *troubleshooting*, siswa akan menggunakan teknik *binary search*. Misalkan di dalam sebuah *directory* terdapat 12 *file* yang belum dapat dipastikan *file* mana yang mengalami *error*. Dari 12 *file* tersebut akan disalin 6 *file* ke dalam *directory* lain, kemudian coba jalankan program kembali. Apabila ketika dijalankan masih *error*, maka dari 6 *file* tersebut akan disalin 3 *file* saja, dan apabila ketika dijalankan masih *error* maka ketiga *file* tersebut akan dicek satu persatu. Teknik ini dianggap jauh lebih efisien dibandingkan apabila seseorang menggunakan *brute-force search* [8].

2.1.5 *Configuration Management and the Cloud*

Configuration Management and the Cloud merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan cara mengimplementasikan proses konfigurasi di dalam *cloud* secara otomatis. Selain itu, siswa juga akan diajarkan cara mengonfigurasi di dalam *cloud*, bagaimana agen *Puppet dan master* berinteraksi satu dengan yang lain, cara menggunakan *Puppet* (sebuah *software* untuk memanajemen konfigurasi di *cloud*) yang sering digunakan dalam dunia industri saat ini [9].

2.1.6 *Automating Real-World Tasks with Python*

Automating Real-World Tasks with Python merupakan kursus yang memiliki 5 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diberikan kasus-kasus yang ada di dalam kursus sebelumnya dan mencoba menyelesaikan proyek akhir. Proyek akhir dibagi ke dalam 4 *mini-project* yaitu memanipulasi *file* gambar dengan menggunakan Python Imaging Library

(PIL), memproses *file* teks menggunakan *Python Dictionary* dan mengunggahnya ke *Running Web Service*, mengotomatisasi PDF dan mengirimkannya melalui email, serta mengotomatisasi informasi katalog [10].

2.2 *Google Data Analytics*

Analisis data merupakan sebuah teknik yang berfokus dalam mendapatkan *insight* (wawasan) terhadap sebuah data. Dengan semakin berkembangnya internet, ukuran penyimpanan yang semakin besar, kemudahan dalam mengakses data, maka teknik ini semakin diminati pada Era Industri 4.0, Data-data yang pada tahun 2005 diperkiraan dalam jumlah *terabytes*, meningkat secara pesat menjadi *petabytes* pada tahun 2010, kemudian menjadi *exabyte* pada tahun 2015 dan menjadi *zettabytes* pada tahun 2020. Analisis data sendiri dapat dibagi menjadi 3 kategori yaitu: analisis deskriptif, analisis prediktif, dan analisis preskriptif [7]. Di dalam Program Bangkit sendiri, terdapat 8 kursus yang dapat dipelajari oleh siswa terkait fondasi-fondasi dalam menganalisis data, mempersiapkan data untuk dieksplorasi, membersihkan data mentah, cara menyampaikan dan memvisualisasikan *insight* (wawasan) dari sekumpulan data, mengimplementasikan analisis data menggunakan Bahasa Pemrograman R, dan di akhir kursus terdapat *capstone project* yang melibatkan kasus di dunia nyata [11].

2.2.1 *Foundations: Data, Data, Everywhere*

Foundations: Data, Data, Everywhere merupakan kursus yang memiliki 5 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan mempelajari darimana data itu berasal; siklus kehidupan data yang biasa disebut dengan SAS; memahami konsep dari identifikasi data, akusisi dan *filtering* data, ekstraksi data, membersihkan dan melakukan validasi data, analisis data, visualisasi data, dan juga mengutilisasi hasil analisis data yang sudah dibuat. Di dalam modul ini juga siswa akan mempelajari sintaks visualisasi data menggunakan *spreadsheet* (seperti mean, median, modus, dll.), dan juga sintaks-sintaks sederhana dari SQL (seperti menggunakan perintah SELECT, WHERE, dll.) [12].

2.2.2 *Ask Questions to Make Data-Driven Decisions*

Ask Questions to Make Data-Driven Decisions merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan mempelajari cara untuk mengajukan pertanyaan dan memecahkan masalah tersebut. Kerangka berpikir dalam mengajukan pertanyaan yang dianjurkan dalam modul ini adalah dengan menggunakan kerangka SMART (*Specific, Measureable, Action-oriented, Relevant, Time-bound*). Hal lainnya yang dibahas dalam modul ini adalah tipe-tipe permasalahan umum yang biasa diselesaikan oleh seorang data analis [13].

2.2.3 *Prepare Data for Exploration*

Prepare Data for Exploration merupakan kursus yang memiliki 5 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan mempelajari cara menghasilkan, mengumpulkan, dan memilih data untuk dianalisis. Pemilihan data dilakukan dengan cara menentukan *time frame* sebelum data analis memutuskan apakah akan mengumpulkan data baru atau menggunakan data yang sudah tersedia, apabila memilih mengumpulkan data baru maka hal yang perlu diperhatikan adalah bagaimana cara mengumpulkan dan berapa banyak data yang perlu dikumpulkan. Sedangkan jika memilih untuk menggunakan data yang sudah tersedia maka hal yang perlu diperhatikan adalah data apa yang akan dipilih. Siswa juga akan diajarkan terkait data terstruktur dan data tidak terstruktur, tipe data, serta format data sebelum akhirnya data tersebut siap untuk dieksplorasi [14].

2.2.4 *Process Data from Dirty to Clean*

Process Data from Dirty to Clean merupakan kursus yang memiliki 6 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya

menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diberikan pembekalan seputar integritas data (di dalamnya termasuk perhitungan statistika, uji hipotesis, dan *margin of error*, ukuran sampel, ukuran bias, sampel acak). Tidak hanya berhenti sampai di situ saja, siswa juga akan diajarkan strategi untuk mengatasi masalah apabila data yang dimiliki tidak mencukupi. Selain itu, data-data tersebut perlu dibersihkan sebelum akhirnya dapat dianalisis [15].

2.2.5 *Analyze Data to Answer Questions*

Analyze Data to Answer Questions merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan mengatur data (*sorting* dan *filter*) sehingga data-data tersebut mudah untuk dianalisis. Proses pengaturan data ini akan dilakukan menggunakan *spreadsheet* dan *SQL*. *Sorting* atau pengurutan data terbagi ke dalam 2 cara yaitu dengan melakukan secara menurun (Z-A) atau menaik (A-Z). *Spreadsheet* juga memiliki fungsi lainnya yang dapat digunakan seperti SORT, SORTBY, dan FILTER [16].

2.2.6 *Share Data Through the Art of Visualization*

Share Data Through the Art of Visualization merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan memahami pentingnya visualisasi data, kunci konsep yang dipakai dalam visualisasi data, dan konsep statistika seperti korelasi dan penyebab (*correlation and causation*). Korelasi di dalam statistika merujuk kepada pengukuran derajat terhadap dua variabel yang berhubungan satu dengan yang lainnya. Misalnya ketika temperatur suhu naik, maka penjualan es krim ikut meningkat ini adalah contoh dari korelasi. Berbeda dengan penyebab, penyebab merujuk kepada sebuah acara (*event*) yang menyebabkan luaran yang spesifik *(specific outcome*) misalnya ketika petir menyambar, maka kita bisa mendengar suara petir tersebut [17].

2.2.7 *Data Analysis with R Programming*

Data Analysis with R Programming merupakan kursus yang memiliki 5 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan penggunaan bahasa pemrograman R dalam membantu proses analisis data. *Environment* yang digunakan dalam mengimplementasikan bahasa pemrograman R adalah dengan Rstudio. Perbedaan mendasar dibandingkan dengan menggunakan Bahasa Python adalah sintaksnya yang menggunakan teknik "*scalpel*" (mencari *packages* yang diinginkan untuk data yang dimiliki) [18].

2.2.8 *Google Data Analytics Capstone: Complete a Case Study*

Google Data Analytics Capstone: Complete a Case Study merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan mencoba mencari studi kasus nyata dan akan dinilai dengan *peer-review*. Di dalam kursus ini juga tersedia forum diskusi untuk didiskusikan bersama dengan rekan lainnya yang mengikuti kursus ini [19].

2.3 *Mathematics for Machine Learning: Linear Algebra*

Mathematics for Machine Learning: Linear Algebra merupakan kursus yang memiliki 5 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan terkait vektor dan matriks. Pada awal kursus, kita akan diberikan intuisi tentang relasi antara kecerdasan mesin, aljabar linear, vektor, dan matriks. Hal ini berguna tidak hanya mencari persamaan yang cocok dalam data yang dimiliki, akan tetapi hal ini juga berguna untuk mencari tahu persamaan yang digunakan tersebut. Cara untuk mengetahui persamaan yang digunakan adalah dengan menemukan nilai optimal parameter yang dapat menyerupai distribusi dari

permasalahan yang sedang dihadapi. Pada kursus ini juga kita akan diajarkan cara menghitung *dot product, modulus, negation* di dalam vektor, mengidentifikasi basis tersebut *linearly independent* atau tidak, mencari determinan dari sebuah matriks, mengeliminasi matriks menggunakan *Gaussian elimination,* mengimplementasikan dalam kode cara mengoperasikan perkalian matriks, dan operasi matriks menggunakan *Einstein Summation Convention,* melakukan refleksi serta manipulasi terhadap gambar, melakukan proses *Gram-Schmidt,* mencari *eigenvector* dan *eigenvalues* dalam memeriksa algoritma *PageRank Google* dama menyajikan hasil pencarian web [20].

2.4 *Mathematics for Machine Learning: Multivariate Calculus*

Mathematics for Machine Learning: Multivariate Calculus merupakan kursus yang memiliki 6 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di awal kursus, siswa akan diberikan intuisi terkait penggunaan kalkulus terhadap kecerdasan mesin. Salah satu fungsi utamanya adalah kalkulus digunakan untuk menganalisis hubungan dari sebuah fungsi terhadap masukan yang diberikan. Kemudian mencoba berlatih cara melakukan perhitungan *sum, product,* dan *chain rules*. Selain itu, siswa juga akan diajarkan cara melakukan diferensiasi terhadap variabel peubah banyak, mengutilisasi struktur vektor/ matriks di dalam kalkulus multivariat, mengimplementasikan *Jacobian*. Setelah memahami konsep dasar dari kalkulus multivariat, maka siswa akan mengimplementasikan dalam bentuk kode cara untuk melakukan *backpropagation* di dalam *neural network* berukuran kecil. Pada modul 4, siswa diajarkan terkait *Taylor Series* dalam upaya menurunkan suatu metode numerik dan juga Matriks Hessian untuk mengoptimasi *function of interest*. Pada modul 5, siswa akan diajarkan cara menemukan *minima* atau *maxima* di dalam *Gradient Descent* (dengan pendekatan *Lagrange multipliers method)*. Di akhir modul, siswa akan diajarkan cara mengimplementasikan fungsi non-linear menggunakan *gradient descent* [21].

2.5 *Mathematics for Machine Learning: Principal Component Analysis*

Mathematics for Machine Learning: Principal Component Analysis merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan dikenalkan terhadap satu algoritma yang sangat bermanfaat dalam kecerdasan mesin yaitu *Principal Component Analysis*. *Principal Component Analysis* digunakan dalam mereduksi dimensi dari sebuah data. Hampir sebagian besar kursus ini berorientasi dalam kode program dimulai dari menghitung dasar statistika (*mean, median, modus*) dari sebuah dataset, menginterpretasikan efek dari transformasi linear, menghitung variansi, mengubah data gambar ke dalam bentuk vektor. Pada modul 2, siswa akan diajarkan cara menghitung dalam bentuk kode program jarak dan sudut antar sebuah gambar, mencari *inner product*, serta menentukan ortogonalitas bergantung pada *inner product*. Di akhir modul, siswa akan diajarkan mengimplementasikan *Principal Component Analysis* walaupun pada modul ini juga dibahas bahwa algoritma ini memiliki pendekatan yang hampir sama dengan *compressing data* mirip seperti .jpg atau .mp3 yang memungkinkan adanya *loss* dari data asli [22].

2.6 *Supervised Machine Learning (Regression and Classification)*

Supervised Machine Learning (Regression and Classification) merupakan kursus yang memiliki 3 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan terkait pengetahuan dasar mengenai *supervised machine learning*. *Supervised machine learning* adalah implementasi pembelajaran mesin yang di dalam datasetnya terdapat fitur dan label. Label tersebut diberikan oleh ahli/ disupervisi oleh manusia. Kelebihan pembelajaran mesin dibandingkan pemrograman tradisional adalah *programmer* tidak perlu menuliskan kode program secara eksplisit, melainkan memasukan data-data tersebut ke dalam sebuah model matematika dan membiarkan model untuk mempelajari karakteristik dari sebuah data. Di dalam kursus ini juga, siswa

diajarkan bagaimana mengimplementasikan *gradient descent*, menghitung *cost function* serta mengoptimasi model regresi menggunakan *gradient descent* [23].

2.7 *Advanced Learning Algorithms*

Advanced Learning Algorithms merupakan kursus yang memiliki 4 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan terkait komponen *neural network*, memahami konsep *layer* di dalam *neural network*, memahami fungsi aktivasi di setiap *layer*, dan mengimplementasi kode dalam *Tensorflow*. Selain itu, siswa juga diajarkan cara mengimplementasi *softmax* untuk melakukan klasifikasi di dalam *neural network*. *Softmax* memberikan *output* berupa probabilitas 0-1, semakin tinggi probabilitas maka semakin *confidence* sebuah model terhadap kelas tersebut [24].

2.8 *Unsupervised Learning, Recommenders, Reinforcement Learning*

Unsupervised Learning, Recommenders, Reinforcement Learning merupakan kursus yang memiliki 3 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan terkait pembelajaran mesin tidak tersupervisi artinya model diberikan sebuah dataset yang tidak memiliki label. Oleh karena itu, model akan berusaha mengklasterisasi karakteristik dari sebuah data. Salah satu algoritma yang digunakan untuk klasterisasi adalah algoritma *k-means*. Siswa akan diajarkan cara menghitung/ *update centroid* di dalam k-means dan mengimplementasi fungsi yang menemukan *centroid* terdekat terhadap sebuah titik [25].

2.9 *Convolutional Neural Network, Natural Language Processing, Time Series*

Convolutional Neural Network, Natural Language Processing, Time Series merupakan 3 kursus utama yang dapat diakses oleh siswa yang sudah terdaftar

untuk memperoleh *TensorFlow Developer Specialization*. Di akhir dari setiap kursus akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan terkait cara memproses data gambar, data teks, maupun data *time series*. Dalam memproses data gambar, siswa juga akan diajarkan cara mengklasifikasi gambar dasar (kucing dan anjing). Selain itu, siswa akan mempelajari cara melakukan *convolutional* dan *pooling*. Sedangkan di kursus *Natural Language Processing* siswa akan diajarkan cara melakukan tokenisasi dalam teks, mengubah data teks menjadi data numerik, serta *pad_sequences APIs* di dalam TensorFlow. Di dalam kelas *Time Series*, siswa akan diajarkan cara menangani data sekuensial yang nilai dari data tersebut berubah seiring waktu seperti temperatur dalam beberapa waktu, jumlah pengunjung di dalam *website*, dan lain sebagainya [26].

2.10 *Structuring Machine Learning Project*

Structuring Machine Learning Project merupakan kursus yang memiliki 2 modul utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap modul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan terkait *streamline* dan optimisasi model *Machine Learning*. Cara memisahkan data *train/dev/test* berdasarkan dataset yang dimiliki. Membandingkan model dengan *human-level performance*. Cara mengimplementasikan *transfer learning*, mengenali *bias*, *variance*, dan ketidakcocokan data dengan melihat performa algoritma pada rangkaian *train/dev/test* [27].

2.11 *Tensorflow Data and Deployment*

Tensorflow Data and Deployment merupakan kursus yang memiliki 4 kursus utama yang dapat diakses oleh siswa yang sudah terdaftar. Di akhir dari setiap mkursusodul akan terdapat ujian praktik maupun teori yang wajib diikuti dalam upaya menyelesaikan kursus dan memperoleh sertifikat. Di dalam kursus ini, siswa akan diajarkan terkait cara *training* model *machine learning* di dalam browser dan menggunakannya dengan menginferensi menggunakan JavaScript. Di dalam

kursus ini juga, siswa akan diajarkan untuk menggunakan *machine learning* secara langsung di browser serta di server backend seperti Node.js [28].

BAB 3
ANALISIS DAN RANCANGAN SISTEM

3.1 Studi Independen (*Machine Learning*)

Studi Independen Jalur Pembelajaran *Machine Learning* merupakan salah satu jalur pembelajaran yang diselenggarakan pada Program Bangkit 2023. Pada jalur pembelajaran ini siswa diajarkan mulai dari logika pemrograman menggunakan *Python*, fondasi matematika terkait *machine learning*, materi pembelajaran mesin baik itu teori maupun praktik, serta materi yang lebih *advance* yaitu teknik-teknik deep learning. Banyaknya kursus yang ditempuh selama mengikuti program ini sebanyak 29 kursus dari Platform Coursera dan 8 kursus dari Platform Dicoding. Rata-rata nilai yang diperoleh selama mengikuti 37 kursus ini yaitu 96.74. Di akhir dari program ini terdapat Sertifikasi Tensorflow yang pada saat penulisan laporan ini masih belum diselenggarakan.

3.2 Teknik Pengumpulan Dataset

Teknik pengumpulan dataset yang digunakan diambil dari data sekunder. Data sekunder tersebut bersumber dari *Kaggle* yang berjudul *Depression: Reddit Dataset* seperti pada tabel 3.1. Data yang digunakan berjumlah 7733 baris dengan nama kolom *'text'* dan *'is_depression'*. Kolom *'text'* merupakan kumpulan data tekstual yang berisikan curahan hati dari konten *Reddit*, dan kolom *'is_depression'* berisikan bilangan biner, apabila data tersebut diisi dengan bilangan 0 maka data tekstual tersebut dikategorikan sebagai data yang tidak memiliki unsur depresi, sedangkan apabila data tersebut diisi dengan bilangan 1 maka data tekstual tersebut dikategorikan sebagai data yang memiliki unsur depresi.

Tabel 3. 1 Dataset Reddit

	text	is_depression
1	we understand that most people who reply immediately to an op with an invitation to talk privately m...	1
…	…	…

7733	slept wonderfully finally tried swatching for new project classic line cardi from stash but ...	0

Selanjutnya data tersebut digabungkan dengan data lainnya yaitu data yang bersumber dari Google Dataset (dirujuk ke pada halaman website *Papers with Code*) dengan judul *SDCNL (Suicide vs Depression Classification)* seperti pada tabel 3.2. Data yang digunakan hanya sebesar 9 baris data dan kolom yang digunakan hanya mengambil kolom *'selftext'* dan *'is_suicide'* dengan *header* yang dibuang kemudian digabungkan dengan *header* milik dataset *Kaggle.*

Tabel 3. 2 Dataset SDCNL

	self_text	**is_suicide**
1	Hi I don't really know how to phrase this situation but I'll try. My life is at a really good point right now, I'm never really depressed ..	1
…	…	…
9	Suicidal people are tired of hearing the same generic shit. Thatâ€™s why we donâ€™t open up	1

Kemudian, dikarenakan 9 baris data tersebut seluruhnya dikategorikan ke dalam kategori depresi maka untuk menyeimbangi dataset tersebut maka ditambahkan dengan satu sumber data lainnya yaitu sumber data yang berasal dari *API* ChatGPT dengan penambahan 8 baris data tekstual yang dikategorikan ke dalam kategori tidak depresi seperti pada tabel 3.3.

Tabel 3. 3 Dataset API ChatGPT

	text	**is_depression**
1	Hey, it's great to connect with you. I'm feeling really good these days and wanted to share a bit about what's been going on in my life. I'm in a really positive place right now	0

	- I have a great job, a supportive social network …	
…	…	…
8	Hey, how's it going? I'm feeling pretty great these days and wanted to share the good vibes. Life has been really good to me lately - I've got a great partner, a fulfilling job, and lots of exciting things ...	0

Total data yang digunakan dalam laporan ini berjumlah 7750 baris data dengan dua kolom yaitu *'text'* dan *'is_depression'*. Akan tetapi, dengan kedua kolom ini saja tidak cukup dikarenakan aplikasi yang ingin dihasilkan dapat diimplementasikan di *region* Indonesia, maka kemungkinan besar input yang akan diterima dalam aplikasi akan berbahasa Indonesia, oleh karena itu pada awal percobaan, dilakukan pembuatan fungsi menggunakan *API* Google Translate dengan cuplikan kode sebagai berikut:

```
def translate_text(text):
  url =
"https://translate.googleapis.com/translate_a/single?client=gtx&sl=en&tl=id&dt
=t&q={}".format(text)
  response = requests.get(url)
  result = response.json()[0][0][0]
return result
```

Gambar 3. 1 Fungsi Translate Text

Fungsi def translate_text(text) yang ditunjukkan pada gambar 3.1 adalah sebuah fungsi untuk melakukan *request* terhadap *API* Google translate dengan *source language* "en" dan *target language* "id". Input parameter yang diterima berupa teks dari dataset kemudian hasil dari fungsi *translate_text* berupa data tekstual yang sudah diterjemahkan ke dalam Bahasa Indonesia. Namun, setelah mencoba mengimplementasikan fungsi ini ke dalam dua skenario yaitu skenario pertama ketika mencoba menggunakan *dummy data* yaitu ketika input data hanya sebanyak 5 baris data saja fungsi ini bisa berjalan dengan baik. Skenario selanjutnya mencoba mengimplementasikan fungsi ini ke dalam dataset yang sesungguhnya,

fungsi *translate_text* ini menyebabkan error pada beberapa kolom baris data yang berisi teks kosong, teks berisi emoji, teks yang bahasanya bercampur (tidak hanya Bahasa Inggris saja).

Dikarenakan untuk menyeleksi dataset secara manual tidak memungkinkan, maka dicoba menggunakan pendekatan lain yaitu dengan mengkonversi data *CSV* tersebut ke dalam format Excel menggunakan Website Convertio, kemudian mengunggah Excel tersebut ke dalam Google Spreadsheet. Setelah dataset berhasil diunggah, penambahan kolom *text_id* dilakukan untuk menyimpan data yang akan diterjemahkan. Penerjemahan yang dilakukan menggunakan fungsi Google Translate yang disediakan di dalam Google Spreadsheet yaitu dengan fungsi =GOOGLETRANSLATE(text, [source_language, target_language]). *Source_language* yang digunakan berupa "en" dan *target_language* yang digunakan berupa "id". Pendekatan kedua ini berhasil untuk menerjemahkan kolom *text* yang berbahasa Inggris walaupun memerlukan waktu yang cukup lama dan hasilnya tidak seakurat apabila menerjemahkan secara manual oleh ahli. Selanjutnya format data Excel tersebut diubah kembali ke dalam format *CSV* menggunakan Website Convertio.

3.3 Analisis Sistem

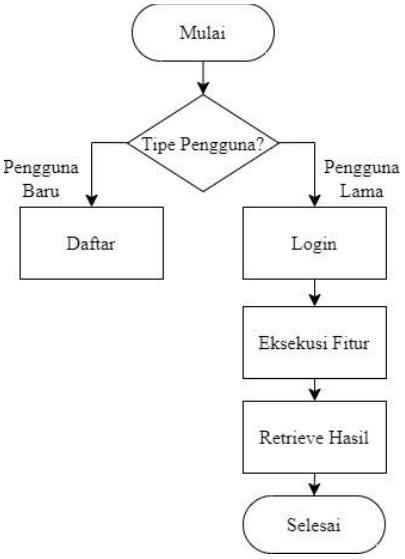
Sistem yang akan dibuat pada *capstone project* berupa sistem yang dapat memantau kesehatan mental penggunanya. Sistem yang dibuat ini akan dikategorikan ke dalam *Human Healthcare and Living Well-beings*. Sistem ini dibangun oleh enam anggota dengan deskripsi pekerjaan sebagai berikut:

Tabel 3. 4 Daftar Anggota

Nama	Path	Asal Universitas	Deskripsi
Sherly Santiadi	Machine Learning	Universitas Kristen Maranatha	Membangun model *deep learning* kategori *natural language processing*
Ahsan Firdaus	Machine Learning	Universitas Gunadarma	Membangun model *deep learning* kategori *computer vision*
Arya Tri Putra	Mobile	Universitas	Membangun aplikasi

Majiah	Development	Kristen Maranatha	android
Kaisar Fredi Valentino	Mobile Development	Institut Teknologi Harapan Bangsa	Membangun aplikasi android
Fahrul Zaman	Cloud Computing	Universitas Bale Bandung	Membangun API dan database
Maya Septiani Br Simbolon	Cloud Computing	Institut Teknologi Harapan Bangsa	Membangun API dan database

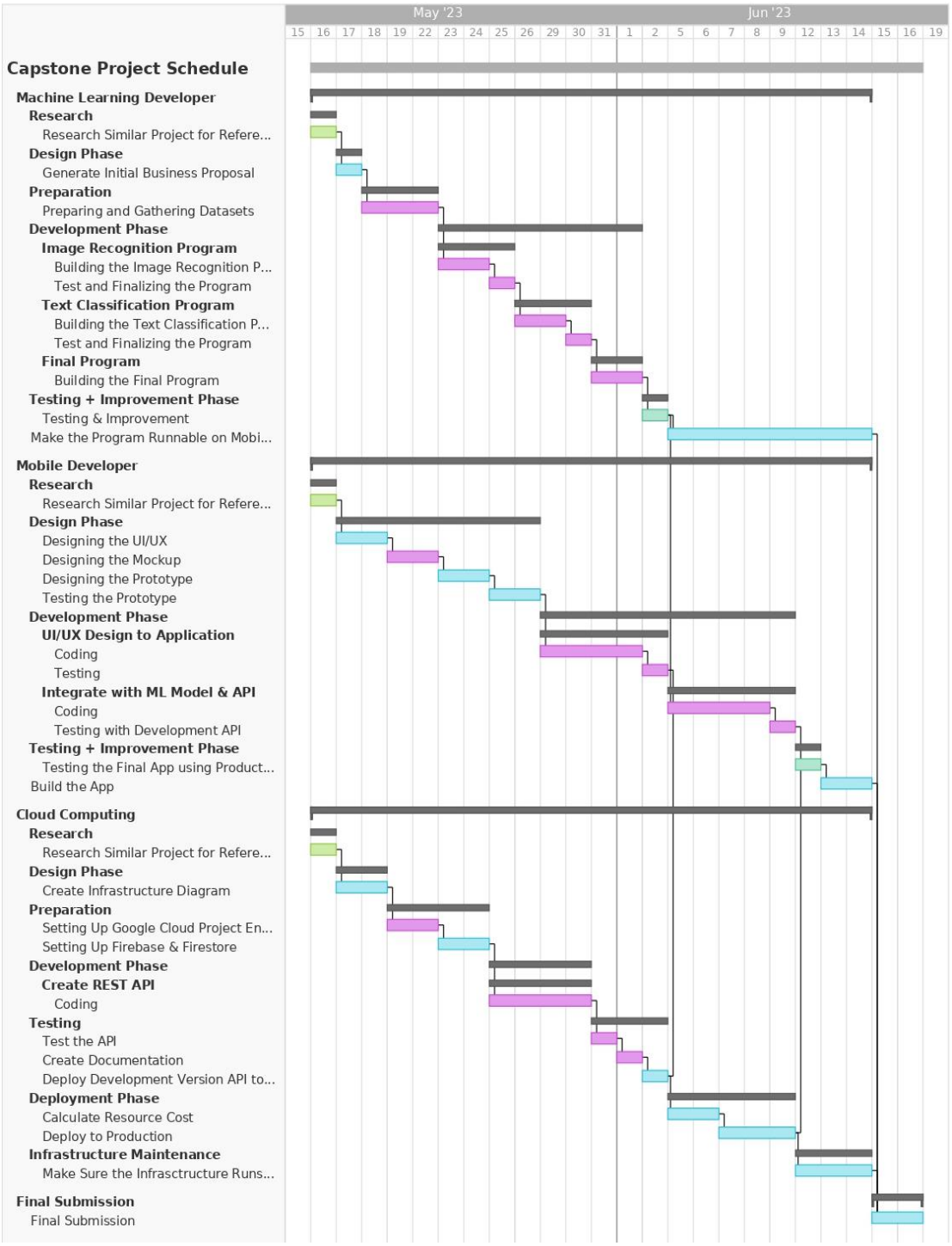
Tujuan dari pembuatan sistem ini adalah untuk mencegah, mengatasi, dan memantau permasalahan seputar kesehatan mental. Di dalam sistem akan terdapat beberapa fitur dengan alur seperti pada gambar 3.2.



Gambar 3. 2 Flowchart Sistem

Pertama, pengguna perlu mendaftarkan diri terlebih dahulu untuk memiliki akun. Jika sudah memiliki akun pengguna dapat login ke dalam aplikasi. Di dalam aplikasi akan terdapat beberapa fitur yang dapat dieksekusi oleh pengguna, kemudian ketika sudah mengeksekusi salah satu fitur, maka pengguna akan mendapatkan hasil yang dihasilkan oleh model *machine learning*.

3.4 Jadwal Proyek



Gambar 3. 3 Jadwal Proyek

Gambar 3.3 merupakan jadwal perencanaan pengerjaan proyek selama satu bulan. Proyek dimulai pada tanggal 16 Mei 2023 dan akan diakhiri pada tanggal 16 Juni 2023. Proyek dimulai dari penyelesaian kursus, pencarian riset maupun jurnal sebagai bahan referensi, pengumpulan proposal bisnis, pengumpulan dataset,

selanjutnya pembangunan model, pembangunan API, pembangunan aplikasi. Proyek akan ditutup setelah model berhasil didaftarkan ke dalam Google Play Store dan presentasi hasil proyek.

3.5 Perancangan Sistem

Terdapat beberapa persyaratan antarmuka eksternal yang diperlukan dalam perancangan sistem agar sistem dapat berjalan dengan baik yaitu:

3.5.1 Antarmuka dengan Pengguna

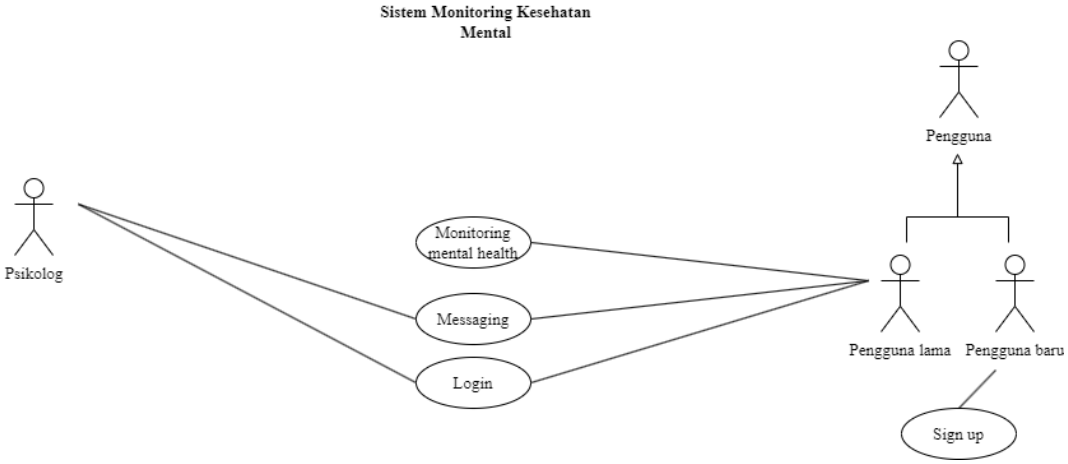
Untuk menggunakan sistem ini, antarmuka dengan pengguna akan ditampilkan di dalam aplikasi Android. Model yang sudah dilatih akan disimpan ke dalam tipe data *Hierarchical Data Format* (*HDF*). Sedangkan untuk data-data yang diperlukan oleh pengguna akan diproses melalui *Google Cloud API*. Visualisasi bahasa yang dipakai pada tampilan sebagian besar menggunakan Bahasa Indonesia dan sedikit perpaduan Bahasa Inggris.

3.5.2 Antarmuka Perangkat Keras

Perangkat *Input/Output (I/O)* yang digunakan yaitu perangkat keras bersistem operasi Android. Aplikasi yang dikembangkan menggunakan Kotlin sehingga tidak akan kompatibel apabila digunakan dalam sistem operasi selain Android. Selain itu, diperlukan juga koneksi internet untuk mengakses fitur-fitur yang tersedia di dalam DailyCloud.

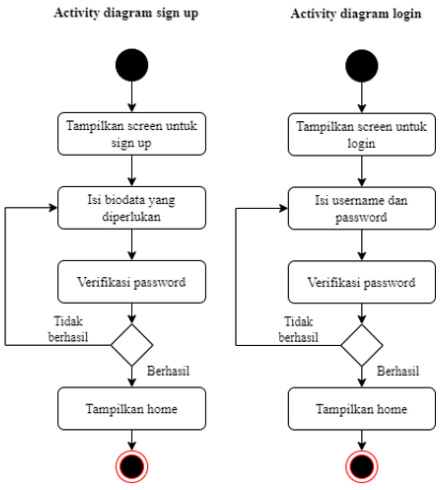
3.5.3 Antarmuka Perangkat Lunak

Sistem perangkat lunak yang dibutuhkan untuk membangun aplikasi dalam laporan ini yaitu *Android Studio* dan *Postman* dengan pustaka *Jetpack Library*, *Retrofit, dan Glide*. Sedangkan untuk konfigurasi di dalam Cloud diperlukan *Visual Studio Code* dengan pustaka *Bootstrap/Tailwind*. Untuk pembangunan model *machine learning* diperlukan *Notebook Kaggle* serta pustaka *Tensorflow*, *Scikit-learn, Numpy*, dan lain-lain. Selain itu terdapat juga beberapa desain UML diantaranya:



Gambar 3. 4 UML Sistem

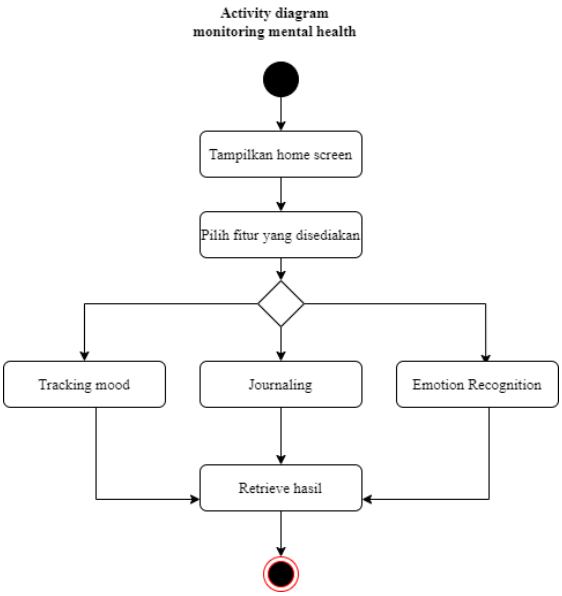
Gambar 3.4 merupakan rancangan desain sistem *monitoring* kesehatan mental. Dalam rancangan ini terdapat 2 aktor yang akan menggunakan sistem *monitoring* kesehatan mental yaitu psikolog dan juga pengguna. Pengguna di dalam sistem dikategorikan ke dalam 2 pengguna yaitu pengguna lama yaitu pengguna yang sudah memiliki *username* dan *password* di dalam aplikasi DailyCloud serta pengguna baru yang perlu mendaftarkan dirinya terlebih dahulu. Dalam sistem monitoring kesehatan mental ini juga terdapat 4 *use cases* yaitu *sign up*, *login*, *messaging*, dan *monitoring* kesehatan mental. Keempat *use cases* tersebut akan dijelaskan lebih detil di dalam *activity diagram*.



Gambar 3. 5 Activity Diagram Sign Up dan Login

Gambar 3.5 merupakan dua buah diagram aktivitas yang menggambarkan alur dari kedua *use cases* yaitu *sign up* dan *login*. Kedua aktivitas ini serupa hanya di dalam aktivitas diagram login pengguna sudah memiliki *username/email* serta

password yang terdaftar sehingga apabila pengguna berhasil masuk ke dalam aplikasi maka pengguna akan langsung dialihkan ke halaman *home*. Sedangkan untuk pengguna yang belum mendaftar belum memiliki *username/email* dan password oleh karena itu pengguna perlu mengisikan kolom-kolom biodata terlebih dahulu sebelum pada akhirnya dapat mengakses halaman *home*.

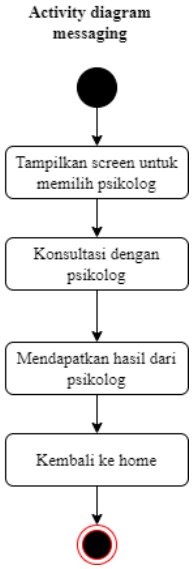


Gambar 3. 6 Activity Diagram Sistem

Gambar 3.6 merupakan diagram aktivitas untuk memantau kesehatan mental. Dalam diagram aktivitas ini diasumsikan bahwa pengguna sudah berhasil *login* ke dalam aplikasi DailyCloud. Pengguna dapat memilih fitur yang disedikaan yaitu *tracking mood*, *journaling*, dan *emotion recognition*. Fitur *tracking mood* digunakan hanya untuk mengisi perasaan apa yang dirasakan *user* hari itu menggunakan stiker yang sudah didefinisikan terlebih dahulu (bahagia, sedih, marah, dan lain-lain).

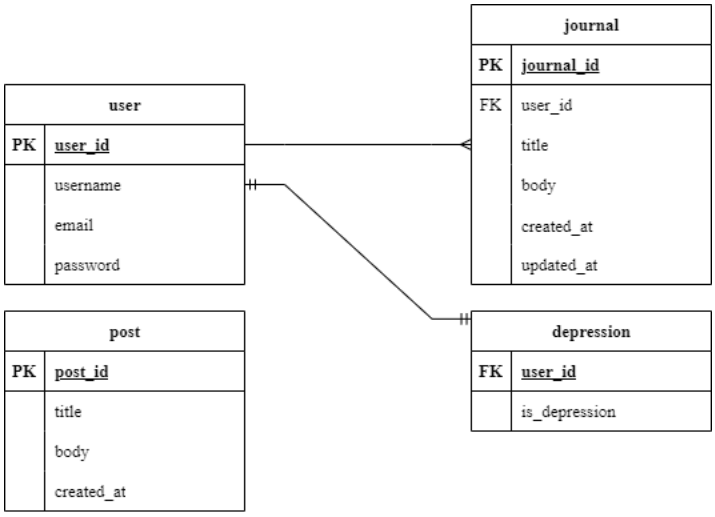
Fitur lainnya yaitu *journaling* merupakan fitur yang sudah ditanamkan model *machine learning* menggunakan metode *supervised learning*. Input yang diterima dalam fitur *journaling* berupa teks dan output yang dihasilkan berupa probabilitas teks tersebut mengarah ke dalam kategori depresi atau tidak. Apabila pengguna terdeteksi depresi oleh sistem, maka akan diarahkan ke dalam fitur *messaging* untuk berkonsultasi langsung dengan psikolog. Selanjutnya terdapat fitur *emotion recognition* yang juga berupa fitur yang sudah ditanamkan model

maching learning berupa *computer vision task*, di dalam fitur tersebut pengguna hanya perlu membuka kamera untuk mendeteksi perasaan berdasarkan raut wajah yang ditampilkan.



Gambar 3. 7 Activity Diagram Messaging

Gambar 3.7 merupakan diagram aktivitas dari fitur *messaging*. Fitur ini disediakan untuk jasa konsultasi langsung dengan psikolog secara *online*. Pengguna yang sudah terdeteksi depresi oleh sistem akan diarahkan ke fitur ini, namun pengguna lainnya yang merasa membutuhkan fitur ini juga dapat langsung berkonsultasi tanpa harus menunggu sistem mendeteksi bahwa pengguna tersebut probabilitas depresinya cukup tinggi. Setelah mendapatkan hasil dari psikolog maka pengguna dapat kembali ke halaman *home* untuk melihat tips seputar cara memanajemen kesehatan mental yang disediakan oleh DailyCloud.



Gambar 3. 8 Desain Penyimpanan Basis Data

Gambar 3.8 merupakan desain penyimpanan basis data. Di dalam aplikasi ini terdapat beberapa tabel yang akan digunakan diantaranya adalah tabel *user, post, journal, depression*. Tabel *user* memiliki kunci primer berupa *user_id* dan tabel ini berelasi terhadap dua tabel lainnya yaitu dengan tabel *journal* dan *depression*. Relasi yang terbentuk antar tabel *user* dan *journal* adalah *one to many* dengan *foreign key* yaitu *user_id*. Sedangkan relasi yang terbentuk antar tabel *user* dan *depression* adalah *one to one* dengan *foreign key* yaitu *user_id*.

3.5.4 Antarmuka Komunikasi

Aplikasi ini merupakan sebuah perangkat lunak yang digunakan untuk mencegah, mengatasi, dan memantau permasalahan seputar kesehatan mental. Perangkat lunak ini berpusat pada hasil prediksi dan analisis kesehatan mental pengguna dan akan menyimpan serta mengirimkan laporan kepada psikolog apabila terdapat keadaan yang dianggap genting untuk segera ditangani oleh ahli. Selain itu, perangkat lunak ini juga mampu menampilkan seputar tips untuk mengelola kesehatan jiwa penggunanya.

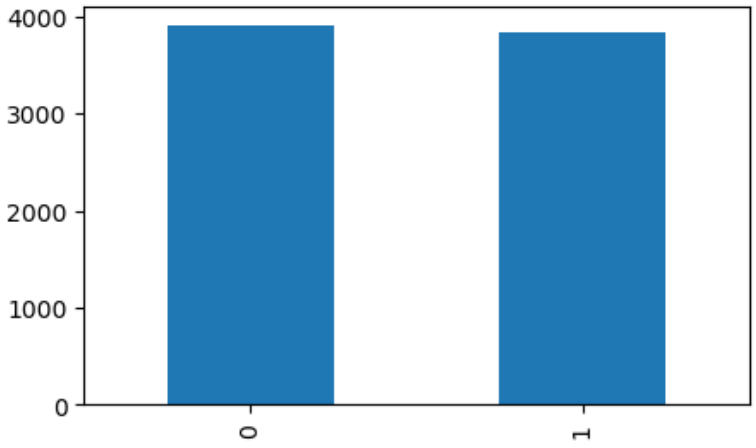
BAB 4
IMPLEMENTASI

4.1 Pembentukan Dataset

Tabel 4. 1 Dataset Laporan

	text	is_depression	text_id
1	we understand that most people who reply immediately to an op with an invitation to talk privately m...	1	Kami memahami bahwa kebanyakan orang yang segera membalas OP dengan undangan untuk berbicara secara ...
2	sleep is my greatest and most comforting escape whenever i wake up these day the literal very first ...	1	Tidur adalah pelarian terbesar dan paling menghibur saya setiap kali saya bangun hari ini, emosi per...
3	i live alone and despite me being prone to loneliness a i find myself to be emotionally needy i seem...	1	Saya hidup sendiri dan meskipun saya rentan terhadap kesepian dan saya mendapati diri saya secara em...
…	…	…	…
7750	need a hug	0	butuh pelukan

Tabel 4.1 merupakan cuplikan data yang digunakan dalam laporan ini dataset yang digunakan memiliki tiga buah kolom yaitu *text, is_depression, text_id*. Kolom *text* berupa data tekstual yang berisikan kalimat yang secara tersirat terkandung makna depresi maupun tidak depresi, sedangkan kolom *text_id* berisikan hal yang sama hanya saja sudah diterjemahkan ke dalam Bahasa Indonesia hal ini dikarenakan *input* data yang nanti akan dibangun di aplikasi Android berupa Bahasa Indonesia, kemudian yang terakhir berupa kolom *is_depression* yang berupa label biner yaitu 0 apabila teks dikategorikan sebagai teks yang bermakna tidak depresi dan 1 apabila teks dikategorikan sebagai teks yang bermakna depresi. Format dataset yang digunakan pada laporan ini yaitu dalam bentuk *Comma Separated Values (CSV)* dengan total jumlah baris data yang digunakan sebanyak 7750 data.



Gambar 4. 1 Visualisasi Label Dataset

Hal yang perlu dipastikan lagi adalah dengan memastikan pada setiap kelas/label yang ada seimbang. Oleh karena itu, dengan menggunakan *bar plot* dua kelas yaitu 0 dan 1 dijumlahkan dan hasilnya adalah data tersebut seimbang seperti pada gambar 4.1. Dengan adanya data yang seimbang maka tidak diperlukan melakukan teknik *oversampling* maupun *undersampling*.

4.2 Pembersihan Data

Sebelum memasukkan data tersebut ke dalam model *machine learning* maka diperlukan beberapa tahap pemrosesan data teks tersebut yaitu dengan cara berikut ini:

In [8]:

```
factory = StemmerFactory()
stemmer = factory.create_stemmer()
stop_words = set(stopwords.words('indonesian'))

### def translate_test lebih baik diskip translate via spreadsheet aja
# def translate_text(text):
#     url = "https://translate.googleapis.com/translate_a/single?client=gtx&sl=en&tl=id&dt=t&q={}".for
mat(text)
#     response = requests.get(url)
#     result = response.json()[0][0][0]
#     return result

def preprocess_text(text):
#     text = translate_text(text) # Kemudian di sini kita coba translate
    text = word_tokenize(text.lower()) # Tokenize text_id ke dalam token/kata
    text = [t for t in text if t not in stop_words] # Hapus stop_words
    text = [stemmer.stem(t) for t in text] # Menjadikan kata dasar
    text = [t if not t.isdigit() else num2words(int(t)) for t in text] # Mengganti angka ke teks
#     text = emoji.demojize(text) # ubah emoji ke dalam bentuk teks
#     text = re.sub(r':[a-z_]+:', lambda m: ' '.join(m.group(0).replace(':', '').split('_')), text) #
Regex misal emoji: 🎉 bakal diubah jadi "party popper".
    text = ' '.join(text) # Gabungkan ke dalam teks kembali
    return text
```

Gambar 4. 2 Pemrosesan Data

Hal pertama yang dilakukan adalah memecah kalimat tersebut menjadi kata-kata atau biasa disebut dengan token. Tokenisasi ini digunakan untuk memampukan model untuk mengekstrak makna dari sebuah *input* yang diberikan. Kemudian setelah mendapatkan token tersebut maka kata-kata yang terlalu sering dimunculkan (diistilahkan dengan *stop words*) lebih baik dihapus dikarenakan biasanya kata-kata tersebut tidak memiliki makna yang berarti bagi model. *Stop words* yang digunakan diambil dari pustaka Sastrawi. Di dalam pustaka Sastrawi terdapat banyak *corpus stop words* yang digunakan seperti kata ‘yang’, ‘untuk’, ‘ke’, ‘para’, dan lain-lain, dapat juga ditambahkan *corpus* baru namun di dalam penulisan laporan ini *corpus* yang digunakan berupa *default corpus* yang disediakan oleh Sastrawi. Kemudian kata-kata yang sudah dibuang melalui *stop words* dilakukan *stemmer* (menjadikan kata sebagai kata dasar) hal ini berguna untuk mengambil *similar word* (kata yang serupa) misalkan saja kata ‘berlari’ akan dijadikan sebagai kata ‘lari’ karena kata ‘berlari’ dan ‘lari’ memiliki makna yang sama hanya saja kata ‘berlari’ sudah diberikan imbuhan. Kemudian data angka akan diubah menjadi tekstual seperti angka ‘8’ misalnya akan dijadikan ‘delapan’ melalui pustaka num2words.

4.3 Proses Data Teks

	text	is_depression	text_id	nlp_text
0	we understand that most people who reply immed...	1	Kami memahami bahwa kebanyakan orang yang sege...	paham banyak orang balas op undang bicara prib...
1	welcome to r depression s check in post a plac...	1	Selamat datang di post Depresi S Posting di te...	selamat post depresi s posting ambil bagi mili...
2	anyone else instead of sleeping more when depr...	1	orang lain daripada lebih banyak tidur ketika ...	orang tidur depresi begadang malam hindar cepa...
3	i ve kind of stuffed around a lot in my life d...	1	Saya telah banyak mengisi banyak dalam hidup s...	isi hidup tunda orang hindar orang dewasa tang...
4	sleep is my greatest and most comforting escap...	1	Tidur adalah pelarian terbesar dan paling meng...	tidur lari besar hibur kali bangun emosi rasa...

Gambar 4. 3 Hasil Pemrosesan Data Teks

Gambar 4.3 merupakan hasil dari pemrosesan dataset melalui fungsi *preprocess_text* dengan input parameter teks. Hasil pemrosesan dataset ini dimasukan ke dalam pandas dataframe dengan nama kolom *'nlp_text'* selanjutnya dari '*nlp_text*' ini dihitung kata unik untuk dimasukan ke dalam sebuah set. Kemudian menggunakan *built-in function* yaitu "*len*" untuk menghitung ukuran set/ *vocabulary* tersebut. Dari hasil dataset yang digunakan ukuran *vocabulary* yang digunakan sebesar 14.233 kata unik.

Selanjutnya dilakukan *one hot encoding* untuk setiap kalimat yang ada di dalam "*nlp_text*" ke dalam daftar indeks kata berukuran 14.233. Kemudian hitung panjang maksimum untuk dimasukan ke dalam *word embeddings*, didapatkan panjang maksimum sebesar 1.300 elemen. Setelah *one hot encoding* diimplementasikan maka *one hot encoding* tersebut perlu dikonversi ke dalam *word embeddings*. Hal ini bermanfaat untuk mengurangi dimensi dan *sparse* yang terjadi ketika mengimplementasikan *one hot encoding*.

Selain itu, pastikan *input* yang dimasukan ke dalam model harus memiliki ukuran yang sama, oleh karena itu perlu dilakukan *padding* menggunakan *pad_sequences*. Parameter yang digunakan dalam laporan ini adalah *padding = 'post', truncating = 'post'*, dengan *maxlen = max_len* yaitu 1.300 elemen. Parameter *padding = 'post'* berfungsi untuk menambahkan *padding* supaya jumlahnya sama dengan *max_len* di akhir. Sedangkan parameter *truncation = 'post'* berfungsi untuk memotong sekuens yang lebih panjang dari *max_len* di akhir.

4.4 Pemisahan Dataset

```
In [19]:
X_train, X_test, y_train, y_test = train_test_split(embedded_docs, y, test_size = 0.1, random_state
= 42, stratify = y)
```

```
In [20]:
# Split data ke dalam training set dan test set (90% training data, 10% test data)
X_train_val, X_test, y_train_val, y_test = train_test_split(embedded_docs, y, test_size=0.1, random_
state=42, stratify=y)

# Split data training set yang tersisa ke dalam training set dan validation set (80% training data, 2
0% validation data)
X_train, X_val, y_train, y_val = train_test_split(X_train_val, y_train_val, test_size=0.2, random_st
ate=42, stratify=y_train_val)
```

Gambar 4. 4 Pemisahan Dataset

Dataset dipecah ke dalam *train-test-val* set sebesar 90% untuk *training data* dan 10% untuk *test data*. *Data training* dipecah kembali menjadi 80% *training data* dan 20% *validation* data untuk melakukan *hyperparameter tuning*. Oleh karena itu, hasil pemisahan dataset menjadi 80% *training data*, 10% *validation data*, dan 10% *test data*.

4.5 Hyperparameter

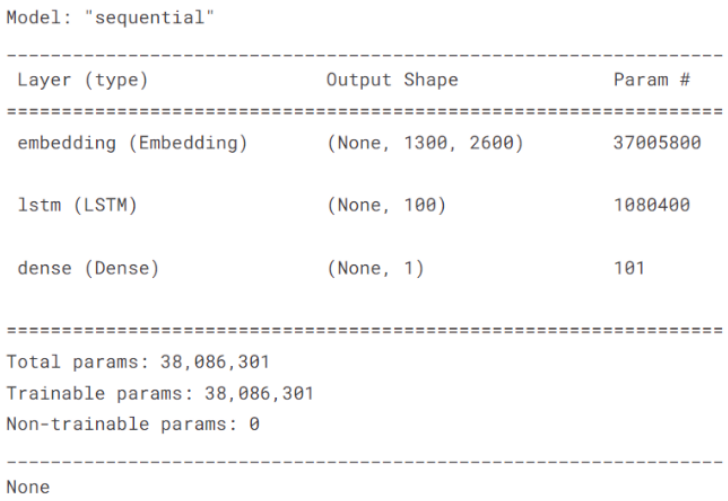
Tabel 4. 2 Hyperparameter

Hyperparameter	LSTM	BiLSTM
Jumlah Layer	1	2
Jumlah Neuron	100	[128,64]
Activation	Sigmoid	Sigmoid
Optimizer	Adam	Adam
Metrics	Accuracy	Accuracy
Loss	Binary Crossentropy	Binary Crossentropy
Return Sequences	False	True
Dropout	None	0.2
Recurrent Dropout	0.2	0.2
Epoch	5	3
Batch Size	16	16

Tabel 4.2 merupakan *best value* yang dihasilkan ketika melakukan *hyperparameter tuning (grid search)*. Jumlah layer pada BiLSTM diperbanyak menjadi 2 layer. Selain itu, jumlah neuron pada LSTM hanya sebanyak 100 neuron saja sedangkan pada BiLSTM jumlah neuron sebanyak 128 (dari *time step* sebelumnya dalam *forward direction*) dan 64 neuron (dari *time step* sesudahnya dalam *backward direction*). Fungsi aktivasi yang digunakan pada kedua model

tetap sama yaitu sigmoid dikarenakan kebutuhan model untuk memprediksi *binary output*. Optimizer yang digunakan pada kedua model juga sama yaitu menggunakan Adam, Adam sendiri dikenal sebagai optimizer yang dapat beradaptasi terhadap *learning rate*. Fungsi loss yang digunakan pada kedua model juga sama yaitu menggunakan binary crossentropy untuk masalah klasifikasi biner. Return sequences pada model BiLSTM disetel True dikarenakan kompleksitas model yang lebih tinggi dibandingkan model LSTM. Kemudian dropout dan recurrent dropout diaplikasikan sebagai parameter regularisasi untuk menghindari *overfitting*. Hal menarik lainnya adalah epoch yang diperlukan oleh BiLSTM lebih sedikit namun memiliki hasil akurasi yang lebih tinggi dibandingkan model LSTM. Terakhir, batch size yang digunakan pada awalnya sebanyak 64 namun operasi sistem pada *Notebook Kaggle* berhenti ketika menangani *batch* sebanyak 64, oleh karena itu *batch size* diturunkan menjadi 16 *batch size*.

4.6 Model Baseline (LSTM)



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1300, 2600)        37005800

 lstm (LSTM)                 (None, 100)               1080400

 dense (Dense)               (None, 1)                 101

=================================================================
Total params: 38,086,301
Trainable params: 38,086,301
Non-trainable params: 0
_________________________________________________________________
None
```

Gambar 4. 5 Model Baseline (LSTM)

Model baseline yang digunakan sebagai perbandingan berupa model LSTM yang dibangun secara sekuensial, dengan input 14.233 sesuai dengan ukuran *vocabulary* yang berarti bahwa model dapat menangani 14.233 kata unik, kemudian jumlah fitur dari setiap kata sebesar *max_len* dikali 2 atau dengan kata lain 1.300 dikali 2 = 2.600. Kemudian ditambah dengan layer LSTM sebesar 100 unit neuron, dan terakhir ditambah dengan dense layer dengan *single output unit* dan fungsi aktivasi sigmoid untuk klasifikasi peluang antara 0 dan 1. Pada model ini

konfigurasi yang digunakan untuk training adalah dengan menggunakan *binary cross entropy* sebagai *loss function*, kemudian *optimizer Adam* dan metriks akurasi sebagai metriks evaluasi.

Sebagai contoh, kita memiliki input dengan sebuah kalimat sederhana "Saya sangat bahagia". Maka, alurnya modelnya akan sebagai berikut:

1. Tokenisasi

   Kalimat tersebut akan ditokenisasi: ["Saya", "sangat", "bahagia"]. Misalkan diasumsikan sudah melalui pemrosesan data teks.

2. Integer Encoding

   Kemudian setiap token akan diberi ID unik seperti 1 untuk "Saya", 2 untuk "sangat", dan 3 untuk "bahagia" maka kalimat "Saya sangat bahagia" menjadi [1,2,3]

3. Padding

   Kemudian diimplementasikan *padding* misalkan dengan *max_len* = 1300 maka menjadi [1,2,3,0,0,0,0,0,0,0,...0] ukuran list yang dihasilkan harus sampai dengan ukuran 1300.

4. Embedding

   Setiap integer ID di dalam *padded sequence* akan dipetakan ke dalam *dense vector* dengan ukuran yang tetap. *Output shape* dari Embedding layer adalah (*batch_size, sequence_length, embedding_dimension*) atau pada laporan ini berupa (None, 1300, 2600). None merepresentasikan *batch size*, 1300 merupakan panjang dari *input sequence*, bilangan 1300 ini didapatkan dari *max_len* dari one hot encoding pada df["nlp_txt"], Dataframe df["nlp_txt"] ini merupakan kolom baru untuk menyimpan teks yang sudah melalui tahap *preprocessing*. Sedangkan bilangan 2600 merupakan panjang vektor embedding dari setiap kata. Jumlah parameter di layer ini adalah 37.005.800, hal ini selaras dengan hasil perkalian *vocabulary* yaitu 14.233 (lihat pada sub bab 4.3 Proses Data Teks) dikalikan dengan panjang vektor *embedding* yaitu 2600. Bilangan 2600 ini merupakan hasil dari penyetelan di awal yaitu 2**max_len* atau 2*1300 = 2600. Vektor *embedding* ini merupakan *hyperparameter* yang akan berisikan vektor berukuran 2600 dari bilangan real. Pada beberapa kali percobaan ketika vektor *embedding* ini

disetel terlalu besar memori *notebook Kaggle* tidak akan cukup untuk memprosesnya.

5. LSTM

   Embedded sequence akan dimasukan ke dalam layer LSTM dengan 100 unit neuron. Jumlah 100 neuron ini berarti dapat menghasilkan vektor berukuran 100 (bisa diartikan sebagai model memiliki 100 *memory cell* berdasarkan *input* dan *previous state*) pada setiap *time step* dalam *input sequence*. Jumlah parameter pada layer ini adalah 1.080.400. Selain itu, LSTM sendiri merupakan variasi dari *Recurrent Neural Network* (RNN) yang dapat menangkap *long-term dependency* di dalam data sekuensial.

6. Dense Layer / Output Layer

   Output dari LSTM akan dimasukan ke dalam *dense layer* dengan sebuah *single unit* beserta fungsi aktivasi sigmoid (memberikan probabilitas 0 dan 1). Jumlah parameter pada layer ini sebesar 101.

7. Loss Function dan Backpropagation

   Output dari model (*predicted y*) kemudian dibandingkan dengan *true label*. *loss* kemudian dilakukan *backpropagate* untuk menyesuaikan bobot untuk memperbaiki performa model. Model akan terus iterasi terhadap keseluruhan training data.

4.7 Model Improvisasi (Bidirectional LSTM)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 1300, 100)         1423300

 bidirectional (Bidirectiona  (None, 1300, 256)        234496
 l)

 bidirectional_1 (Bidirectio  (None, 128)              164352
 nal)

 dense (Dense)               (None, 1)                 129

=================================================================
Total params: 1,822,277
Trainable params: 1,822,277
Non-trainable params: 0
_________________________________________________________________
None
```

Gambar 4. 6 Model Improvisasi (Bidirectional LSTM)

Berbeda dengan model *baseline*, untuk model kedua yang dibangun menggunakan model *Bidirectional LSTM*. Walaupun dengan matriks *embedding* yang jauh lebih kecil yaitu sebesar 100 fitur per vektor dikarenakan memori yang digunakan tidak cukup apabila disetel terlalu besar, namun dengan pembangunan model yang lebih *powerful* diharapkan akan menghasilkan hasil prediksi yang lebih baik. Kemudian penambahan parameter *return_sequences* yang berarti bahwa layer akan mengembalikan urutan output untuk setiap input *time step* (bukan hanya output akhir) dan juga penambahan parameter *dropout* dan *reccurent_dropout* sebesar 20% untuk menghindari *overfitting*. Perbedaan mendasar antara *dropout* dan *reccurent_dropout* adalah *dropout* digunakan untuk menghindari *overfitting* pada model untuk mengingat *training set* sedangkan *recurrent_dropout* digunakan untuk menghindari *overfitting* terhadap *time sequence* tertentu.

Untuk lebih jelasnya, kita memiliki input dengan sebuah kalimat sederhana “Saya sangat bahagia”. Maka, alurnya modelnya akan sebagai berikut:

1. Tokenisasi

   Kalimat tersebut akan ditokenisasi: [“Saya”, “sangat”, “bahagia”]. Misalkan diasumsikan sudah melalui pemrosesan data teks.

2. Integer Encoding

Kemudian setiap token akan diberi ID unik seperti 1 untuk “Saya”, 2 untuk “sangat”, dan 3 untuk “bahagia” maka kalimat “Saya sangat bahagia” menjadi [1,2,3]

3. Padding

   Kemudian diimplementasikan *padding* misalkan dengan *max_len* = 1300 maka menjadi [1,2,3,0,0,0,0,0,0,0,...0] ukuran list yang dihasilkan harus sampai dengan ukuran 1300.

4. Embedding

   Setiap integer ID di dalam *padded sequence* akan dipetakan ke dalam *dense vector* dengan ukuran yang tetap. *Output shape* dari Embedding layer adalah (*batch_size*, *sequence_length, embedding_dimension*) atau pada laporan ini berupa (None, 1300, 100). None merepresentasikan *batch size*, 1300 merupakan panjang dari *input sequence*, bilangan 1300 ini didapatkan dari *max_len* dari one hot encoding pada df[“nlp_txt”], Dataframe df[“nlp_txt”] ini merupakan kolom baru untuk menyimpan teks yang sudah melalui tahap *preprocessing*. Sedangkan bilangan 100 merupakan panjang vektor embedding dari setiap kata (pemilihan *embedding_dimension* ini berbeda dengan model LSTM sebelumnya karena ketika dimensi disetel terlalu besar memori *notebook Kaggle* tidak akan cukup untuk memprosesnya.

5. Bidirectional LSTM (layer 1)

   Embedded sequence akan dimasukkan ke dalam layer Bidirectional LSTM layer pertama dengan 128 unit. Penggunaan bidirectional LSTM ini secara konseptual dapat membantu jaringan untuk mempelajari konteks dari masa lalu serta masa yang akan datang. Perbedaan yang paling mendasar antara model LSTM dengan model Bidirectional LSTM adalah misalkan ketika menggunakan LSTM model akan mempelajari urutan dari kiri ke kanan, dimulai dari kata “saya”, kemudian “sangat”, dan “bahagia”. Sedangkan, pada model *bidirectional LSTM* proses akan dilakukan bukan hanya dari kiri ke kanan, namun akan dipelajari dari urutan kanan ke kiri yaitu dengan mempelajari urutan yang dimulai dari kata “saya”, kemudian “sangat”, dan “bahagia” dan juga dimulai dari kata “bahagia”, kemudian “sangat”, dan “saya”. Dengan mengimplementasikan ini, model bisa mempelajari kata

yang akan muncul setelah kata yang ditunjuk, misalkan ketika memproses kata “saya” maka *backward* LSTM akan memproses kata “bahagia”. Sama halnya ketika memproses kata “bahagia” maka *forward* LSTM akan memproses kata “saya”. Dengan hal ini, model akan bisa menangkap baik dari konteks yang lalu dan juga konteks yang akan datang.

6. Bidirectional LSTM (layer 2)

   Layer ini memproses *sequence vector* dari kedua arah, dengan output *single* vektor sebesar 128 yang merepresentasikan seluruh *sequence* (jadi bukan lagi *sequences of vectors* melainkan menjadi *single* vektor yang merepresentasikan seluruh *sequence* misalnya di dataset kita ada “saya sangat bahagia”, “saya suka membaca”, dan lain sebagainya maka keseluruhan dataset inilah yang diperhatikan dibandingkan hanya memperhatikan “saya sangat bahagia” saja). Cara untuk melakukannya adalah dengan menyetel *hyperparameter ‘return_sequences’* menjadi *False*. Hal ini berguna ketika kita ingin mengklasifikasi *sequence* dari keseluruhan arti dibandingkan dengan elemen individual dari *sequence* yang ada.

7. Dense Layer / Output Layer

   Output dari LSTM akan dimasukkan ke dalam *dense layer* dengan sebuah *single unit* beserta fungsi aktivasi sigmoid (memberikan probabilitas 0 dan 1). Jumlah parameter pada layer ini sebesar 129.

8. Loss Function dan Backpropagation

   Output dari model (*predicted y*) kemudian dibandingkan dengan *true label*. *loss* kemudian dilakukan *backpropagate* untuk menyesuaikan bobot untuk memperbaiki performa model. Model akan terus iterasi terhadap keseluruhan training data.

BAB 5
PENGUJIAN

5.1 *Train-Validation-Test* (80/10/10 *Rule*)

```
model.fit(X_train,y_train,validation_data=(X_val,y_val),epochs=3,batch_size=
16)
model.fit(X_train,y_train,validation_data=(X_test,y_test),epochs=3,batch_size=
16)
```

Gambar 5. 1 Snippet Code Train-Validation-Test

Gambar 5.1 merupakan cuplikan kode untuk model mempelajari data pelatihan beserta labelnya. Selanjutnya model akan mencoba memprediksi terhadap data validasi serta data pengujian. Pemisahan data validasi dengan data pengujian ini bertujuan agar tidak terjadi kebocoran informasi data ketika melakukan *hyperparameter tuning*. Parameter lainnya yang perlu diatur adalah parameter *epoch*, parameter ini berfungsi untuk melakukan iterasi terhadap seluruh kumpulan data selama pelatihan model. Selama satu *epoch*, model menerima seluruh 16 *data point* sekaligus, melakukan *forward propagation* melalui *network*, menghitung *loss*, lalu melakukan *backpropagation gradient* untuk memperbarui bobot model. Proses ini diulangi untuk setiap batch yang terdiri dari 16 *data point* hingga seluruh data training telah diproses. Pada laporan ini terdapat 7750 dataset dengan komposisi sebagai berikut:

```
Shape of X train is: (5580, 1300)
Shape of y train is: (5580,)
Shape of X val is: (1395, 1300)
Shape of y val is: (1395,)
Shape of X test is: (775, 1300)
Shape of y test is: (775,)
```

Gambar 5. 2 Komposisi Data

Maka, ketika melakukan *fitting model* dengan *batch_size* 16, terdapat 7750 / 16 = 484.375 *steps*. Pemilihan *batch_size* ini merupakan hasil percobaan dari kelipatan 8, apabila *batch_size* disetel terlalu besar *GPU* yang digunakan berhenti karena kehabisan memori.

5.2 *Classification Report*

	precision	recall	f1-score	support
0	0.80	1.00	0.89	805
1	1.00	0.73	0.84	742
accuracy			0.87	1547
macro avg	0.90	0.86	0.86	1547
weighted avg	0.90	0.87	0.87	1547

Gambar 5. 3 Classification Report Baseline Model (LSTM)

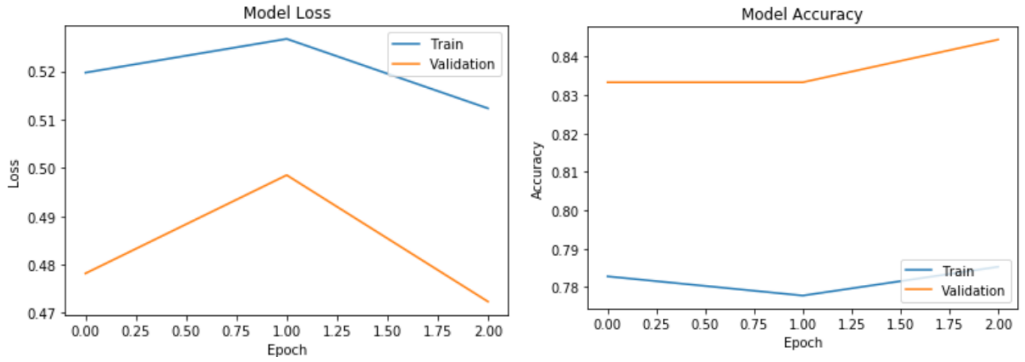
Hasil klasifikasi dari model *baseline* mendapatkan hasil akurasi sebesar 87% dengan tingkat *f1-score* 89% pada label 0 (tidak depresi) dan 84% pada label 1 (depresi). Hasil klasifikasi ini masih tergolong baik namun diharapkan bisa ditingkatkan menjadi lebih baik lagi dengan model *bidirectional LSTM*.

	precision	recall	f1-score	support
0	0.92	0.96	0.94	390
1	0.96	0.92	0.94	384
accuracy			0.94	774
macro avg	0.94	0.94	0.94	774
weighted avg	0.94	0.94	0.94	774

Gambar 5. 4 Classification Report Model Improvisasi (Bidirectional LSTM)

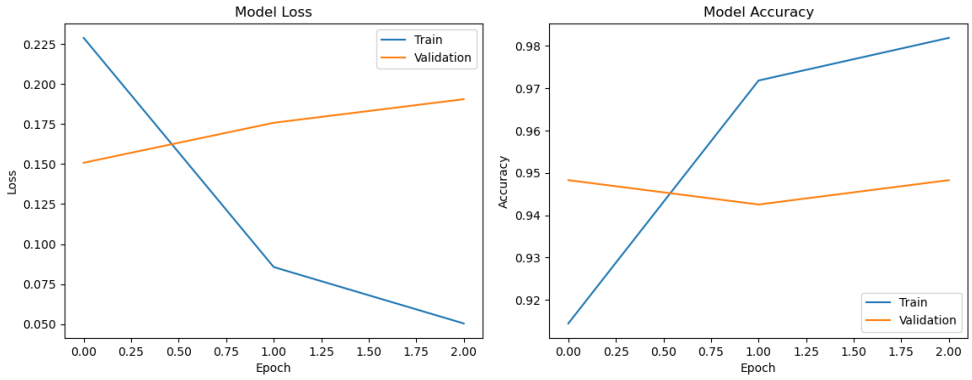
Hasil klasifikasi dari model *baseline* mendapatkan hasil akurasi sebesar 94% dengan tingkat *f1-score* 94% pada label 0 (tidak depresi) dan 94% pada label 1 (depresi). Tingkat akurasi naik sebesar 7% dibandingkan dengan model *baseline*.

5.3 *Plot Model Loss* dan *Accuracy*



Gambar 5. 5 Plot LSTM Model Loss dan Accuracy

Hasil *plot* model *loss* pada gambar 5.5 dapat dilihat bahwa pada *validation* semakin besar *epoch* maka *loss* yang dihasilkan juga semakin besar. Begitu pula pada *plot* akurasi di *validation* terlihat pada *epoch* 0 sampai dengan 1 terdapat penurunan akurasi namun dari 1 menuju *epoch* 2 semakin meningkat walaupun tetap berada pada tingkat akurasi sebesar 0.84.



Gambar 5. 6 Plot Bidirectional LSTM Model Loss dan Accuracy

Hasil *plot* model *loss* pada gambar 5.6 dapat dilihat bahwa pada *validation* semakin besar *epoch* maka *loss* yang dihasilkan juga semakin besar. Begitu pula pada *plot* akurasi di *validation* terlihat pada *epoch* 0 sampai dengan 1 terdapat penurunan akurasi namun dari 1 menuju *epoch* 2 semakin meningkat walaupun tetap berada pada tingkat akurasi sebesar 0.95.

5.4 Konversi Model TF Lite

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.target_spec.supported_ops = [tf.lite.OpsSet.TFLITE_BUILTINS,
tf.lite.OpsSet.SELECT_TF_OPS]
converter._experimental_lower_tensor_list_ops = False
tf_lite_model = converter.convert()
tflite_model_file = pathlib.Path("/kaggle/working/model.tflite")
tflite_model_file.write_bytes(tf_lite_model)
```

Gambar 5. 7 Konversi Model TF Lite

Gambar 5.7 merupakan cuplikan kode untuk mengubah model ke dalam format *TF Lite*. Dengan pengubahan format ini memungkinkan model untuk diimplementasikan ke dalam aplikasi *mobile* atau juga menggunakan model ini untuk melakukan *testing* selain dari dataset yang dimiliki. Untuk melakukan konversi hal pertama yang perlu dilakukan adalah mengimpor pustaka Keras kemudian memanggil *method* TFLiteCoverter. Pustaka lainnya yang perlu diimpor adalah pustaka Path untuk memberikan alamat model akan disimpan. Dikarenakan pada laporan ini *notebook* yang digunakan berupa *Kaggle* maka model yang sudah dikonversi akan disimpan ke dalam *working path* "/kaggle/working/model.tflite".

5.5 *Confidence Testing*

Pada *confidence testing*, beberapa kalimat dimasukan ke dalam model yang sudah dikonversi menjadi tipe data HDF. Berikut merupakan contoh teks yang dimasukan ke dalam model:

```
data = pd.DataFrame([
"Hari ini, saya bangun pagi-pagi sekali karena ingin menyelesaikan beberapa tugas pekerjaan yang belum selesai. Saya merasa cukup lelah karena semalam sempat begadang hingga pukul 2 pagi. Setelah sarapan, saya mulai bekerja dan fokus pada tugas-tugas yang harus saya selesaikan hari ini. Saya berhasil menyelesaikan beberapa tugas, namun ada satu tugas yang memakan waktu lebih lama dari yang saya perkirakan."], columns=['text'])
```

Gambar 5. 8 Data Testing ke-1

Kemudian gambar 5.9 merupakan hasil *preprocessing* data teks dan kemudian disimpan ke dalam *dataframe* 'nlp_text'.

	text	nlp_text
0	Hari ini, saya bangun pagi-pagi sekali karena ingin menyelesaikan beberapa tugas pekerjaan yang belum selesai. Saya merasa cukup lelah karena semalam sempat begadang hingga pukul 2 pagi. Setelah sarapan, saya mulai bekerja dan fokus pada tugas-tugas yang harus saya selesaikan hari ini. Saya berhasil menyelesaikan beberapa tugas, namun ada satu tugas yang memakan waktu lebih lama dari yang saya perkirakan.	bangun pagi selesai tugas kerja selesai lelah malam begadang two pagi sarap fokus tugas selesai hasil selesai tugas tugas makan kira

Gambar 5. 9 Hasil Preprocessing Data Testing ke-1

Gambar 5.10 merupakan hasil dari prediksi model yang menyatakan bahwa teks tersebut dikategorikan kepada kelas 0 (tidak depresi).

```
y_pred = (y_pred >= 0.5).astype("int")
if(y_pred==0):
    print("tidak depresi")
elif(y_pred==1):
    print("depresi")
```

```
tidak depresi
```

Gambar 5. 10 Hasil Prediksi Data Testing ke-1

Kemudian dicoba lagi dengan teks lainnya, berikut merupakan contoh teks yang dimasukan ke dalam model:

```
data = pd.DataFrame([
"Hari ini rasanya begitu berat. Saya merasa tidak ada semangat untuk melakukan apa pun. Saya terus merasa sedih dan lelah meskipun saya tidak tahu persis apa yang membuat saya merasa seperti itu. Saya mencoba untuk menjaga diri saya tetap sibuk dengan bekerja atau melakukan aktivitas lain yang saya sukai, tetapi kadang-kadang itu tidak cukup membantu. Saya merasa sangat kesepian dan terisolasi, meskipun saya tahu bahwa saya memiliki orang-orang yang peduli dengan saya. Saya hanya berharap bahwa suatu hari saya akan bisa merasa lebih baik dan merasa lebih bahagia lagi."], columns=['text'])
```

Gambar 5. 11 Data Testing ke-2

Kemudian gambar 5.12 merupakan hasil *preprocessing* data teks dan kemudian disimpan ke dalam *dataframe* ‘nlp_text’.

	text	nlp_text
0	Hari ini rasanya begitu berat. Saya merasa tidak ada semangat untuk melakukan apa pun. Saya terus merasa sedih dan lelah meskipun saya tidak tahu persis apa yang membuat saya merasa seperti itu. Saya mencoba untuk menjaga diri saya tetap sibuk dengan bekerja atau melakukan aktivitas lain yang saya sukai, tetapi kadang-kadang itu tidak cukup membantu. Saya merasa sangat kesepian dan terisolasi, meskipun saya tahu bahwa saya memiliki orang-orang yang peduli dengan saya. Saya hanya berharap bahwa suatu hari saya akan bisa merasa lebih baik dan merasa lebih bahagia lagi.	berat semangat sedih lelah persis coba jaga sibuk aktivitas suka kadang bantu sepi isolasi milik orang peduli harap bahagia

Gambar 5. 12 Hasil Preprocessing Data Testing ke-2

Gambar 5.13 merupakan hasil dari prediksi model yang menyatakan bahwa teks tersebut dikategorikan kepada kelas 1 (depresi).

[168]:
```
y_pred = (y_pred >= 0.5).astype("int")
if(y_pred==0):
    print("tidak depresi")
elif(y_pred==1):
    print("depresi")
```

```
depresi
```

Gambar 5. 13 Hasil Prediksi Data Testing ke-2

BAB 6
SIMPULAN DAN SARAN

6.1 Simpulan

Wawasan terkait teori maupun praktik *machine learning* yang diberikan selama Program Bangkit 2023 ini dapat diimplementasikan secara nyata ke dalam sebuah proyek akhir. Proyek akhir yang diambil dalam penulisan laporan ini berkaitan langsung dengan topik *Human Healthcare and Living Well-beings*. Adapun tujuan dari proyek akhir ini untuk menciptakan aplikasi yang dapat mencegah, memantau, dan mengatasi problematika terkait kesehatan mental bagi pengguna aplikasi DailyCloud. Dalam proyek akhir ini juga terdapat beberapa kesimpulan yang dapat disimpulkan yaitu:

- Hasil visualisasi data yang digunakan menunjukan bahwa dataset yang digunakan dalam laporan ini termasuk ke dalam kategori seimbang. Selain itu, *raw dataset* yang pertama kali digunakan menggunakan Bahasa Inggris yang kemudian diterjemahkan ke dalam Bahasa Indonesia. Namun, apabila dilihat secara mendetil struktur *raw dataset* yang dimiliki cukup berantakan dan terkadang tidak memiliki makna sama sekali. Selain itu, jumlah dataset yang dapat diolah menggunakan *notebook* Kaggle sangat terbatas.
- Implementasi algoritma *deep learning* yang dihasilkan berupa model *Bidirectional LSTM*. Model ini merupakan improvisasi dari model *baseline* yang dipakai sebelumnya yaitu model *LSTM*.
- Akurasi yang dihasilkan sebesar 94% dan dapat dikategorikan cukup baik walaupun masih terdapat *room of improvement*. Pemilihan metriks ini dikarenakan dataset yang digunakan seimbang.

6.2 Saran

Walaupun hasil akurasi yang diperoleh oleh model *Bidirectional LSTM* sudah cukup baik dibandingkan model *baseline*, namun masih terdapat *room of improvement* yang dikembangkan lagi agar model menjadi lebih baik lagi dalam melakukan *sentiment analysis*. Berikut merupakan saran untuk pengembangan aplikasi di masa yang akan datang:

- Membersihkan *raw dataset* terlebih dahulu sebelum mengimplementasikan *preprocessing* teks.
- Menggunakan dataset berbahasa Indonesia.
- Menambah jumlah dataset yang lebih bervariasi.
- Menambah ukuran *layer* maupun *neuron*.
- Menggunakan model yang lebih *powerful* seperti *Transformers*.

DAFTAR PUSTAKA

[1] M. Tohir, “Buku Panduan Merdeka Belajar - Kampus Merdeka,” 2020, doi: 10.31219/osf.io/ujmte.

[2] Y. Hendayana, “Program Bangkit : Kolaborasi Kampus Merdeka dengan Google, Gojek, Tokopedia, Traveloka Resmi dimulai Hari Ini,” 2021. https://dikti.kemdikbud.go.id/kabar-dikti/kabar/program-bangkit-kolaborasi-kampus-merdeka-dengan-google-gojek-tokopedia-traveloka-resmi-dimulai-hari-ini/ (accessed Apr. 03, 2023).

[3] Dicoding, “About,” 2015. https://www.dicoding.com/about (accessed Apr. 03, 2023).

[4] A. Sweigart, *Automate The Boring Stuff*. 2015.

[5] Google, “Crash Course on Python,” 2012. https://www.coursera.org/learn/python-crash-course/home/welcome (accessed Apr. 01, 2023).

[6] Google, “Using Python to Interact with the Operating System,” 2012. https://www.coursera.org/learn/python-operating-system/home/welcome (accessed Apr. 01, 2023).

[7] Google, “Introduction to Git and GitHub,” 2012. https://www.coursera.org/learn/introduction-git-github/home/welcome (accessed Apr. 02, 2023).

[8] Google, “Troubleshooting and Debugging Techniques,” 2012. https://www.coursera.org/learn/troubleshooting-debugging-techniques/home/welcome (accessed Apr. 01, 2023).

[9] Google, “Configuration Management and the Cloud,” 2012. https://www.coursera.org/learn/configuration-management-cloud/home/module/1 (accessed Apr. 02, 2023).

[10] Google, “Automating Real-World Tasks with Python,” 2012. https://www.coursera.org/learn/automating-real-world-tasks-python/home/welcome (accessed Apr. 01, 2023).

[11] Google, “Google Data Analytics Professional Certificate,” 2012. https://www.coursera.org/professional-certificates/google-data-analytics

(accessed Apr. 02, 2023).

[12] Google, “Foundations: Data, Data, Everywhere,” 2012. https://www.coursera.org/learn/foundations-data/home/module/1 (accessed Apr. 02, 2023).

[13] Google, “Ask Questions to Make Data-Driven Decisions,” 2012. https://www.coursera.org/learn/ask-questions-make-decisions/home/welcome (accessed Apr. 02, 2023).

[14] Google, “Prepare Data for Exploration,” 2012. https://www.coursera.org/learn/data-preparation/home/welcome (accessed Apr. 02, 2023).

[15] Google, “Process Data from Dirty to Clean,” 2012. https://www.coursera.org/learn/process-data/home/module/5 (accessed Apr. 03, 2023).

[16] Google, “Analyze Data to Answer Questions,” 2012. https://www.coursera.org/learn/analyze-data/home/module/1 (accessed Apr. 03, 2023).

[17] Google, “Share Data Through the Art of Visualization,” 2012. https://www.coursera.org/learn/visualize-data/home/module/1 (accessed Apr. 03, 2023).

[18] Google, “Data Analysis with R Programming,” 2012. https://www.coursera.org/learn/data-analysis-r/home/module/1 (accessed Apr. 03, 2023).

[19] Google, “Google Data Analytics Capstone: Complete a Case Study,” 2012. https://www.coursera.org/learn/google-data-analytics-capstone/home/module/1 (accessed Apr. 03, 2023).

[20] Imperial College London, “Mathematics for Machine Learning: Linear Algebra,” 2012. https://www.coursera.org/learn/linear-algebra-machine-learning/home/module/1 (accessed Apr. 13, 2023).

[21] Imperial College London, “Mathematics for Machine Learning: Multivariate Calculus,” 2012. https://www.coursera.org/learn/multivariate-calculus-machine-learning/home/module/1 (accessed Apr. 13, 2023).

[22] Imperial College London, “Mathematics for Machine Learning: PCA,”

2012. https://www.coursera.org/learn/pca-machine-learning/home/module/1 (accessed Apr. 13, 2023).

[23] DeepLearning.AI, “Supervised Machine Learning: Regression and Classification,” 2012. https://www.coursera.org/learn/machine-learning/home/module/1 (accessed Apr. 17, 2023).

[24] DeepLearning.AI, “Advanced Learning Algorithms,” 2012. https://www.coursera.org/learn/advanced-learning-algorithms/home/module/1 (accessed Apr. 20, 2023).

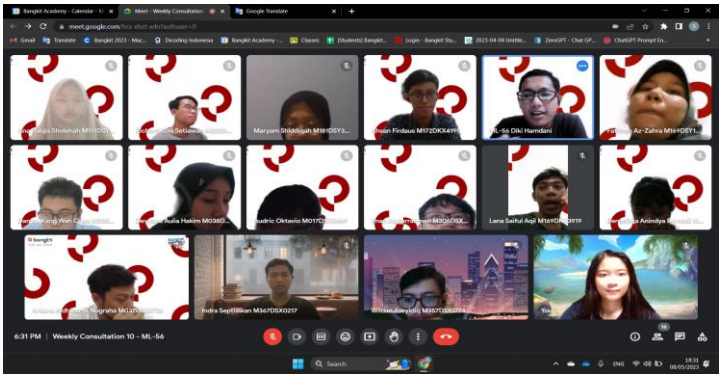
[25] DeepLearning.AI, “Unsupervised Learning, Recommenders, Reinforcement Learning,” 2012. https://www.coursera.org/learn/unsupervised-learning-recommenders-reinforcement-learning/home/module/1 (accessed Apr. 17, 2023).

[26] DeepLearning.AI, “TensorFlow Developer Professional Certificate,” 2012. https://www.coursera.org/professional-certificates/tensorflow-in-practice (accessed Apr. 17, 2023).

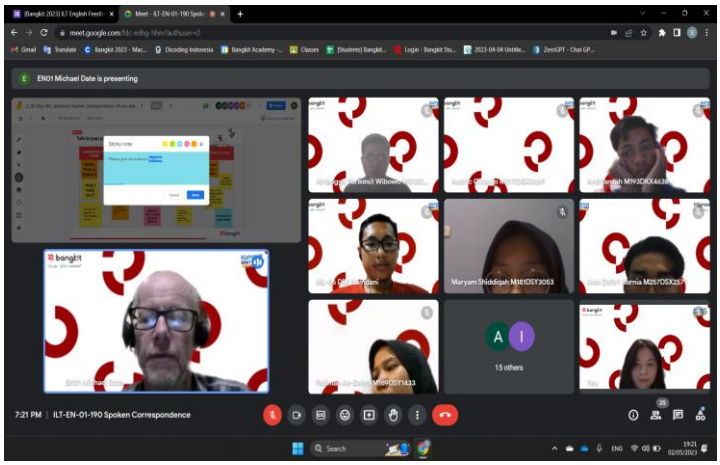
[27] DeepLearning.AI, “Structuring Machine Learning Projects,” 2012. https://www.coursera.org/learn/machine-learning-projects/home/module/1 (accessed Apr. 21, 2023).

[28] DeepLearning.AI, “TensorFlow: Data and Deployment Specialization,” 2012. https://www.coursera.org/specializations/tensorflow-data-and-deployment (accessed Apr. 22, 2023).

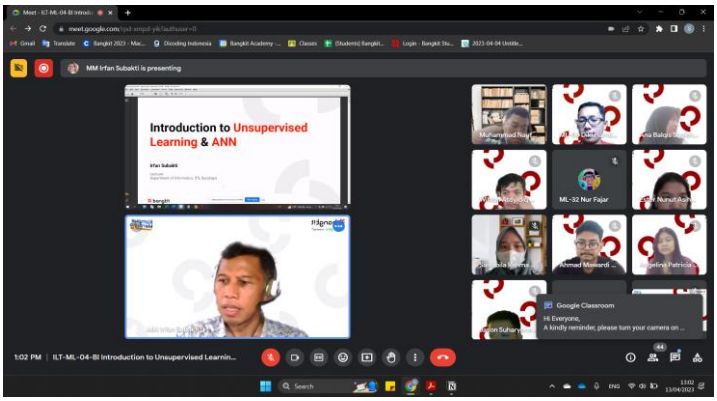
LAMPIRAN A KEGIATAN



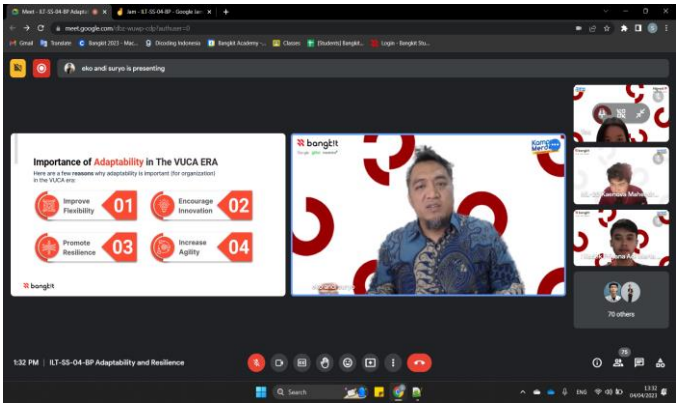
Gambar Lampiran A. 1 Weekly Consultation



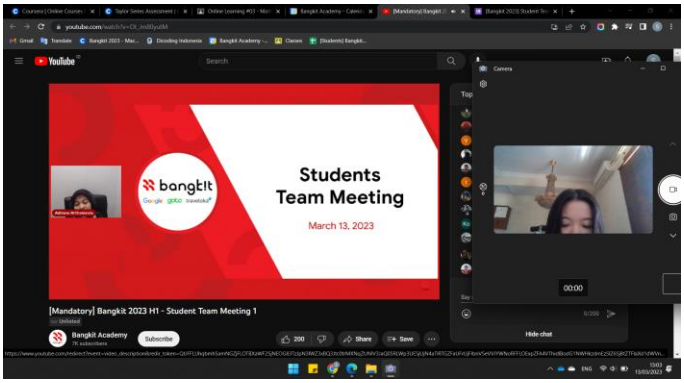
Gambar Lampiran A. 2 Instructor-Led Training (English Class)



Gambar Lampiran A. 3 Instructor-Led Training (Technical Class)



Gambar Lampiran A. 4 Instructor-Led Training (Soft-skills Class)



Gambar Lampiran A. 5 Student Team Meeting

RIWAYAT HIDUP PENULIS

Sherly Santiadi lahir di Kota Bandung dan merupakan anak kedua dari dua bersaudara. Penulis pernah menempuh pendidikan di Sekolah Menengah Atas Swasta Katolik Santa Angela jurusan Matematika dan Ilmu Alam. Selanjutnya, penulis melanjutkan pendidikan di Program Studi Teknik Informatika Universitas Kristen Maranatha. Selama perkuliahan berlangsung, penulis juga terlibat aktif dalam mengikuti berbagai organisasi salah satunya yaitu penulis pernah menjabat sebagai Sekretaris Administrasi dalam organisasi Senat Mahasiswa Fakultas Teknologi Informasi 2022/2023 dan menjadi bagian dari Duta Maranatha 2022/2023. Untuk mengasah keterampilannya, penulis juga aktif dalam mengikuti berbagai riset dosen dan kompetisi. Penulis juga pernah meraih prestasi diantaranya: Juara III *Website Programming Competition* di Universitas Pakuan, Juara I *Start-Up Competition 1.0* Kategori *Business Plan* di Fakultas Bisnis Universitas Kristen Maranatha, Juara I Kategori *Tools* pada Kompetisi Pembelajaran Daring Inovatif Universitas Kristen Maranatha, dan juga pernah menjabat sebagai *Scranton Essay Contest Awardee* yang diselenggarakan oleh *Scranton Women's Leadership Center* di Korea.